National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

UNIVERSITY OF ALBERTA

THE SPATIAL DEVELOPMENT OF THE MAGNETOSPHERIC LOW-LATITUDE BOUNDARY LAYER

BY

C

JOHN R. MANUEL

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Department of Physics

Edmonton, Alberta

Fall 1992

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-77418-5

Canada

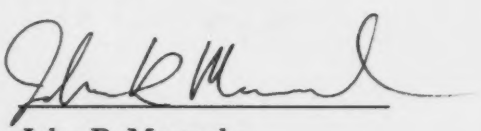
UNIVERSITY OF ALBERTA

RELEASE FORM

Name of Author:	John R. Manuel
Title of Thesis:	The Spatial Development of the Magnetospheric Low-Latitude Boundary Layer
Degree:	Doctor of Philosophy
Year This Degree Granted:	1992

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.



John R. Manuel
Department of Physics
University of Alberta
Edmonton, Alberta
T6G 2J1
Canada

9 October 1992

Perhaps the most valuable result of all education is the ability to make yourself do the thing you have to do, when it ought to be done.

T. H. Huxley (1825-1895)

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled "The Spatial Development of the Magnetospheric Low-Latitude Boundary Layer" submitted by John R. Manuel in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Physics.

J. C. Samson, Supervisor

A. N. Kamal

R. L. Lysak

G. Rostoker

W. Rozmus

G. E. Swaters

9 October 1992

ABSTRACT

The low-latitude boundary layer (LLBL) comprises a large fraction of the magnetospheric boundary layer making it a potentially important site for transport of mass, momentum and energy from the high-speed magnetosheath plasma into the magnetosphere. I have examined, by computer simulation, the processes involved in the spatial development of a 6.4 R_E (Earth radii) long section of the dayside LLBL from a thin and laminar boundary layer to a broad and turbulent one capable of significant transport. The computer simulation developed for this purpose is based on the full set of ideal magnetohydrodynamic (MHD) equations that govern the dynamics of most magnetospheric plasmas and uses a two-dimensional nonperiodic simulation geometry to permit the realistic downstream development of the boundary layer.

Simulations started from several realistic initial conditions all exhibit the formation of a LLBL that broadens with downstream distance, from an upstream thickness of 0.12 R_E to as much as ~0.7 R_E downstream, and reproduces many of the observed boundary layer characteristics. The broadening occurs through the action of Reynolds and Maxwell stresses generated by the Kelvin-Helmholtz (KH) instability in the boundary layer which deposit momentum and energy into the LLBL. The KH instability also transports mass into the LLBL by mixing plasma across the boundary layer through continuous vortex roll-ups and mergings and also appears capable of aiding diffusive transport processes by steepening density gradients at the magnetopause enough to trigger any of a number of possible diffusion processes.

Simulations have also shown that the downstream development of the boundary layer may be slowed and possibly stopped in the presence of a flow-aligned component of the magnetosheath magnetic field. For example, for a magnetosheath magnetic field which is initialized to tilt 30° away from perpendicular to the flow, the KH instability still develops, but fails to generate the

large, merging vortices necessary to the success of the transport processes because the vortices lose kinetic energy to magnetic field distortions as they wind up the magnetic field. The magnetosheath magnetic field is thus capable of nonlinearly stabilizing the KH instability in the LLBL even though the LLBL is linearly unstable to the KH instability.

ACKNOWLEDGMENTS

I wish to thank my supervisor, John Samson, for his unfailing support and the latitude he gave me in choosing my sometimes meandering research direction. Both he and Gordon Rostoker have proven to be excellent guides into the jungle of the Earth's magnetosphere.

Thanks are due to Martin Connors for helping me to realize a long-standing interest in fluid dynamics computer simulations by introducing me to the ALFVEN computer code, Eric Donovan for the magnetic field lines used in Figure 1 and Przemek Frycz for the analytical results used in Figure 5. I also thank my fellow graduate students for many discussions, useful and otherwise, and their friendship: Catherine de Groot-Hedlin, Barry Harrold, Wanda Kamocki, Stephen Kidd, Eric Donovan, Jeff Candy, Terry Kolber, Brian Jackel, Susan Skone and Martin Connors. I owe a special thanks to Gillian Wiley: her support and friendship helped to make this thesis possible.

I would also like to thank the Natural Sciences and Engineering Research Council, the Province of Alberta and the University of Alberta for supporting me through my graduate program.

TABLE OF CONTENTS

Introduction 1

Solar Wind-Magnetosphere Coupling 4

Formation and Dynamics of the LLBL 7

Objectives 11

Linear Stability 13

Simulation 16

Simulation Boundary Layer 24

Numerical Results and Discussion 30

Case B 30

Cases A, B and C 46

Conclusions 63

Bibliography 69

Appendix A 79

Appendix B 80

Introduction 80

Single Step Algorithms 82

Two Step Algorithms 88

Test Problems 95

Test Results and Analysis 98

Discussion and Conclusions 107

Appendix C 110

LIST OF TABLES

Table

1 Parameters of the LLBL crossing 27

2 Parameters and phase velocities of the KH instability for the three boundary layer configurations 29

B.1 Diffusion, antidiffusion and prediffusion coefficient sets for the predictor-corrector formulation of Book and Fry's extension to ETBFCT given in (B.19)-(B.24) 93

LIST OF FIGURES

Figure

1 The outer magnetosphere and magnetosheath 2

2 Magnetic merging of regions of antiparallel magnetic fields 5

3 The development of the KH instability in a thin shear layer [after *Batchelor*, 1967] 8

4 The spatial growth rate of the KH instability found by a linearized perturbation analysis for a realistic boundary layer configuration 15

5 The density field some time after introducing a perturbation at the bottom center of the frame 18

6 The temporal growth of the KH instability found by simulation and predicted by a linear perturbation analysis for a realistic boundary layer configuration 19

7 The plasma and magnetic field parameters observed during the inbound ISEE-1 magnetopause crossing of 17 August 1978 [from *Eastman et al.*, 1985] 25

8 A snapshot taken at $t = 1.0$ minute of a temporally developing shear layer started from the boundary layer configuration of case B 32

9 The compression wave field ($\nabla \cdot \mathbf{v}$) generated by the noisy magnetosheath inflow boundary 33

10 The spatial growth of the KH instability for three boundary layer configurations 34

11 Snapshots of the three simulated LLBL configurations taken at $t = 3.4$ minutes 36

12 Snapshots of streaklines for the three simulated LLBL configurations .42

13 $|\nabla\rho|$ at $t = 3.4$ minutes for boundary layer configuration of case B 45

14 Snapshots of streaklines for the three simulated LLBL configurations .49

15 Time-averaged boundary layer for the three simulated LLBL configurations 50

16 Cross-section of the average shear layer along the line plotted in Figure 15 for case B 54

17 Time-averaged shear stresses and energy flux for the three simulated LLBL configurations 59

18 A model of the spatial development of the LLBL based on the three simulated LLBL configurations 67

B.1 A semicircular density profile after being convected by three different algorithms 81

B.2 The exact amplitude and phase error of the single step ETBFCT algorithm for three sets of coefficients 86

B.3 The exact amplitude and phase error of the corrector step of the predictor-corrector ETBFCT algorithm for three sets of coefficients 94

B.4 A semicircular density profile after being convected 36 cells by the predictor-corrector form of the original ETBFCT algorithm for a range of Courant numbers 99

B.5 A semicircular density profile after being convected 36 cells by the predictor-corrector form of Book and Fry's extended ETBFCT algorithm with this appendix's tunable coefficients for a range of Courant numbers 101

B.6 The one dimensional shock tube at $t = 6.496 \times 10^{-3}$ for the predictor-corrector form of the original ETBFCT algorithm 103

B.7 The one dimensional shock tube at $t = 6.496 \times 10^{-3}$ for the predictor-corrector form of Book and Fry's extended ETBFCT algorithm with this appendix's tunable coefficients 104

B.8 Density in the nonlinear stage of the KH instability using Boris's coefficient set and this appendix's tunable set 105

LIST OF SYMBOLS AND ABBREVIATIONS[1]

a_{LLBL}	half-width of LLBL
a_{MP}	half-width of magnetopause
$\mathbf{B}$	magnetic field
$\mathbf{B}_{\mathrm{gen}}$	generated magnetic field
c	generic wave speed
c_{A}	Alfvén speed
c_{F}	magnetosonic speed
c_{S}	Sound speed
f	one of ρ, $\rho\mathbf{v}$, $\mathbf{B}$ and p
δf	f perturbation
HLBL	high-latitude boundary layer
IMF	interplanetary magnetic field
in (subscript)	inflow boundary value
KH	Kelvin-Helmholtz
k	wavenumber
k_{max}	most unstable wavenumber
l	length scale
LLBL	low-latitude boundary layer
LT	Local Time
M_{C}	convective Mach number
M_{F}	magnetosonic Mach number
max (subscript)	maximum value
min (subscript)	minimum value
MHD	magnetohydrodynamic(s)
MP	magnetopause
n	number density

[1] Excluding those used in Appendix B.

out (subscript)	outflow boundary value
p	pressure
δp	pressure perturbation
p_{amb}	ambient pressure
r	radial coordinate
r_i	ion cyclotron radius
R_E	Earth's radius (6371.2 km)
T_{avg}	averaging period
t	time
Δt	numerical time step
U_B	magnetic field energy
U_{tot}	total plasma energy
UT	Universal Time
v	velocity
v_{ph}	phase speed
x	downstream coordinate of simulation
y	cross-stream coordinate of simulation
z	vertical coordinate of simulation
0 (subscript)	background value
1 (subscript)	magnetospheric value (except where noted)
2 (subscript)	magnetosheath value (except where noted)
β	ratio of fluid and magnetic pressures
γ	adiabatic index (taken as 5/3)
δ	relaxation constant
δ_{ij}	Kronecker delta
η_{ano}	anomalous viscosity
θ	angular coordinate
μ_0	permeability of free space
ν_{ano}	anomalous kinematic viscosity

ρ	mass density
τ	relaxation time
τ_{ij}	Maxwell stress tensor
ϕ and α_F	magnetic field direction
Ω	angular velocity
ω	angular frequency
ω_{max}	most unstable angular frequency

INTRODUCTION

All celestial bodies, ranging in size from asteroids to galaxies, are thought to be surrounded by a tenuous but very energetic and electrically active gas-like medium called a plasma. Some of these bodies possess a magnetic field which pervades the adjacent plasma and communicates the presence of the body to the plasma through a variety of electromagnetic and plasma processes. The region of surrounding plasma having a magnetic field that may be traced back to the body is known as the body's magnetosphere[2].

The magnetosphere of a body situated in an infinite volume of unmagnetized, stationary plasma fills all space since its magnetic field fills all space. In reality, however, magnetospheres are often confined by the presence of a surrounding magnetic field and the existence of plasma motions directed toward the body supporting the magnetosphere. In most space plasmas, the magnetic field behaves as if it is frozen into the plasma and follows the plasma motions [e.g., *Parks*, 1991] and in this way, the "upwind" regions of a magnetosphere are pushed back toward the body supporting them. How much the magnetosphere is compressed against the body depends on the speed of the flow and the strength of the magnetic field. For example, the Earth's magnetosphere is situated in a high-speed stream of Sun-generated plasma called the solar wind. The speed of the solar wind is great enough that the dynamic pressure associated with it balances the magnetospheric magnetic field pressure at a distance of approximately 10 Earth radii (R_E) upstream of the Earth. At times of exceptionally high solar wind speeds, the solar wind dynamic pressure can be large enough to push the subsolar point of magnetosphere to within 5 R_E of the Earth before the Earth's nearly dipolar magnetic field is strong enough to balance it.

The action of the solar wind pressure confines the terrestrial magnetosphere

---

[2] This definition is strictly true only for a closed magnetosphere. Some regions of the Earth's magnetosphere actually appear to be permanently open to the surrounding plasma environment.

to a cavity. The cavity is ellipsoidal in shape at its sunward side [e.g., *Spreiter et al.*, 1966; *Sibeck et al.*, 1991] and is stretched into a long tail on its downstream side as shown in Figure 1. At the downstream side, the deflected solar wind plasma exerts a confining dynamic pressure on the magnetosphere which weakens with downstream distance as the interface between it and the solar wind, the magnetopause, becomes increasingly aligned with the free-stream solar wind velocity vector. The magnetopause stresses that arise from solar wind-magnetosphere interactions [*Axford and Hines*, 1961] keep the magnetosphere from closing under the influence of the other components of the solar wind pressure and stretch it into a tail-like structure, the magnetotail, which extends for at least 240 R_E downstream [*Tsurutani and von Rosenvinge*, 1984].

Before the solar wind reaches the magnetopause, it passes through a shock, the bow shock, created by the obstruction presented to the supermagnetosonic (i.e., moving faster than fast magnetosonic mode waves in the plasma) solar wind by the magnetosphere. As in supersonic gas dynamics, information about the obstacle is carried by waves which propagate only a finite distance upstream

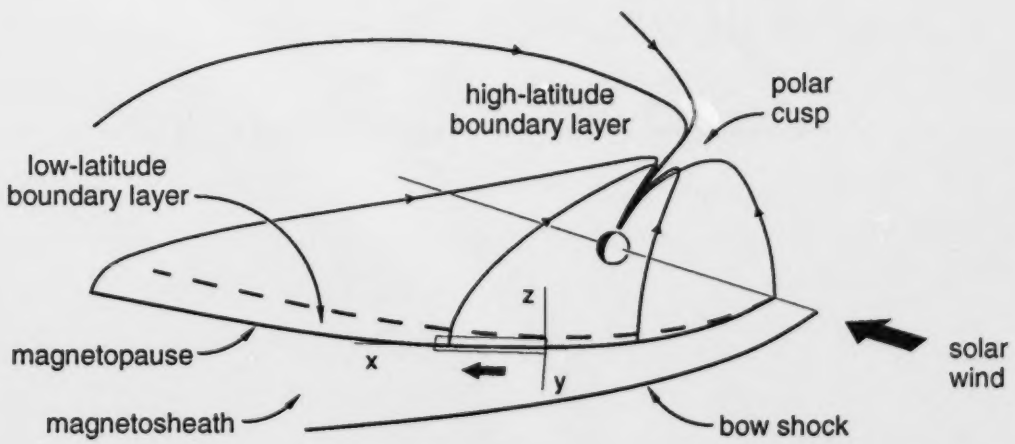


Figure 1. The outer magnetosphere and magnetosheath. The magnetopause and field lines plotted come from an empirical magnetosphere model [*Tsyganenko*, 1989] with $K_p = 0$ and an untilted 30,500 nT R_E^3 dipole.

before encountering a flow faster than the wave speed. The waves "pile up" at that point and form a shock front across which the plasma slows, becoming denser and hotter and often experiencing a change in the fluid pressure and magnetic field [*Spreiter et al.*, 1966]. The solar wind properties change sufficiently across the shock to merit a new name for the region of post-shock solar wind plasma: the magnetosheath.

The magnetosheath region completely surrounds the magnetosphere and is the source of energy for almost all magnetospheric processes. The solar wind, and thus the magnetosheath, carries in the vicinity of 10^{13} J of kinetic, thermodynamic and magnetic energy to the magnetosphere every second (see Appendix A). Some fraction of that energy is transported through the boundary and into the magnetosphere where it becomes available to drive magnetospheric processes. The size of the fraction is determined by the ability of processes occurring at the magnetospheric boundary to couple the solar wind to the magnetosphere.

The topology of the magnetospheric magnetic field leads to a natural division of the magnetospheric boundary into three regions (Figure 1). The smallest of the regions is the polar cusp. Though the cusp accounts for very little of the total boundary area, its magnetic field is connected directly to the magnetosheath thus providing a direct route for entry of magnetosheath plasma into the magnetosphere. The cusp is bounded on all sides by field lines which extend into the two other regions. The magnetic field lines at the tailward side of the cusp connect to the high-latitude boundary layer (HLBL) and remain open to the solar wind. Those at the sunward side of the cusp thread the low-latitude boundary layer (LLBL) and continue to the cusp in the opposite hemisphere while remaining closed to the solar wind. All of these regions contribute to the transfer of energy from the solar wind to the magnetosphere.

SOLAR WIND-MAGNETOSPHERE COUPLING

The boundary layers found at and just inside the magnetopause—the only magnetospheric boundary layers that I consider in this thesis—all contain a tailward-moving plasma with characteristics intermediate between those of the magnetosheath and magnetosphere [e.g., *Lundin*, 1988]. The magnetosheath plasma is dense and cool relative to the adjacent magnetosphere plasma and moves tailward at high speeds [*Hones et al.*, 1972] while the magnetosphere plasma moves slowly and unsteadily sunward [e.g., *Mitchell et al.*, 1990]. The existence of a boundary layer of finite thickness between these two regions indicates that solar wind mass, momentum and energy manage to couple into the magnetosphere through boundary layer processes [*Eastman et al.*, 1976].

It is possible for energy to be transported into the magnetosphere without requiring the operation of any boundary layer processes. Magnetohydrodynamic (MHD) waves generated in the solar wind and magnetosheath are partially transmitted through the magnetopause and carry a significant amount of energy into the magnetosphere [*Verzariu*, 1973]. Once there, they may drive processes which trap the energy in the magnetosphere [e.g., *Goertz and Smith*, 1989; *Samson et al.*, 1992].

There are three basic categories of processes capable of transporting the mass, momentum and energy required to produce the boundary layers observed: magnetic merging of magnetosheath and magnetosphere magnetic fields, impulsive penetration of magnetosheath plasmas and diffusive mixing at the magnetopause. All three categories are likely to be represented at the magnetopause, but the large variation in plasma parameters and magnetic field over the magnetopause could lead to regions where one of the boundary forming processes is preferred over the others.

Magnetic merging occurs when two regions of plasma having antiparallel magnetic fields are driven together by external forces. When the magnetic field

between the two regions is very small and plasma conductivity is finite, the field will no longer be frozen to the plasma and field lines will be able to rearrange in ways that are not possible when bound to a continuous fluid. In particular, the field lines in the merging zone can change their topology and link formerly distinct regions of plasma as shown in Figure 2. Such reconfigurations of the magnetic fields introduce pressure gradient and magnetic field "tension" forces which drive plasma out along the boundary between the two regions and make continuous merging possible [e.g., *Shivamoggi*, 1985].

Magnetic merging is commonly thought to occur at the magnetopause in regions where the magnetosheath flow drives magnetosheath magnetic fields against antiparallel magnetospheric fields [*Dungey*, 1961]. Because the orientation of the magnetosheath magnetic field is continually changing, merging is expected to occur at different patches on the magnetopause at different times. For example, when the interplanetary magnetic field (IMF) carried by the solar wind is directed southward with respect to the Earth's geomagnetic axis, it should merge at the subsolar region of the magnetopause while a northward-directed IMF may be able to merge at the sunward edge of the HLBL near the cusp. At a merging region, the magnetosheath plasma would gain access to the magnetosphere through the recently merged field lines which connect the two

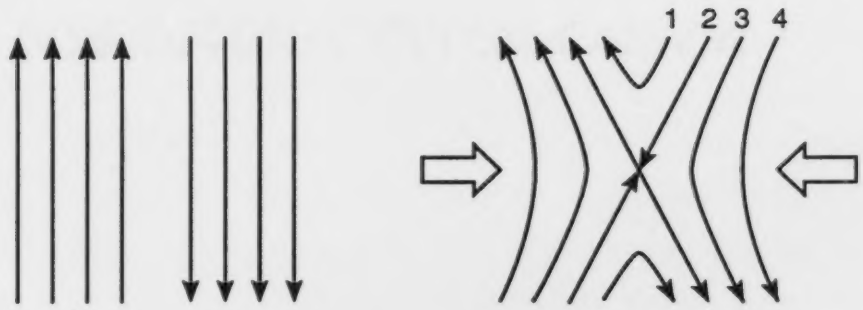


Figure 2. Magnetic merging of regions of antiparallel magnetic fields. Initially parallel fields are driven together and merge where the field becomes very weak. The field line labelled as "1" has merged with a field line from the other side of the merging layer and now connects the two formerly distinct regions.

regions. Dense and cool magnetosheath plasma would then flow along such field lines to form a boundary layer inside the magnetopause [*Russell and Elphic*, 1979]. In this scenario, the magnetic field threading the two regions aids in transporting momentum and energy between them: the fast-moving magnetosheath plasma carries the field lines tailward and they in turn drag the lighter magnetospheric plasma leading to a net transfer of momentum and energy into the magnetosphere.

Impulsive penetration of irregularities in the solar wind plasma through the magnetopause is another possible transport mechanism [*Lemaire and Roth*, 1978; *Roth*, 1992]. Observations have shown [*Celnikier et al.*, 1987] that the solar wind momentum density varies over distances much shorter than the diameter of the magnetosphere. These regions of higher momentum density are deflected less than the background magnetosheath flow deflected around the magnetosphere and may have sufficient momentum density to be able to reach and penetrate the magnetopause and partially populate the boundary layer with magnetosheath-like plasma.

The last category of transport mechanism, and the one this thesis is concerned with, relies on diffusive processes to carry mass, momentum and energy into the magnetosphere. In conventional fluid dynamics, the presence of viscous molecular diffusion causes a velocity shear layer to thicken, resulting in transport across the layer. The molecular collisions that lead to diffusion in conventional fluid dynamics are far too infrequent in magnetospheric plasmas to be able to account for the amount of magnetosheath plasma seen in the boundary layer. However, wave-particle interactions that occur between small-scale plasma turbulence and particles gyrating in response to the Lorentz force scatter the particles in much the same way as Coulomb collisions would and are believed to broaden the boundary layer enough to account for the observed magnetopause thickness [*Tsurutani and Thorne*, 1982], but not the observed thickness of the LLBL [*LaBelle and Treumann*, 1988]. Simulations of the large-scale MHD

turbulence created by the Kelvin-Helmholtz (KH) instability in the LLBL show that the instability can broaden the boundary layer sufficiently to account for the observed thickness [*Miura*, 1984]. Although the physical details of these processes are quite different from classical molecular diffusion, both may be described as diffusive processes [e.g., *Ichimaru*, 1973; *McComb*, 1990].

FORMATION AND DYNAMICS OF THE LLBL

The LLBL "forms" in the sense that it is observed to broaden with distance from the subsolar point [*Eastman and Hones*, 1979] where the boundary layer is thin, when present [*Paschmann et al.*, 1978]. Tailward broadening of the boundary layer is consistent with the operation of a viscous diffusion process which acts to smooth the sharp profiles often seen during satellite passes of the layer [*Sonnerup*, 1980]. However, the profiles observed are not smooth enough to be attributed solely to the action of a diffusive process (Figure 7). The detail seen during satellite passes appears to be due to spatial and temporal variations within the LLBL itself rather irregular motions of the entire LLBL past the observing spacecraft [*Sckopke et al.*, 1981], so any process invoked to describe the formation of the LLBL must be capable of producing such variations.

The KH instability is capable of both broadening an initially thin velocity shear layer and generating small-scale, time-varying structures in the resulting boundary layer. The instability develops out of the free energy present in a shear flow by amplifying small displacements of the shear layer. The vorticity composing a perturbed shear layer protrudes into the ambient shear flows and is swept by them into regions of enhanced vorticity which further perturb the boundary layer (Figure 3) and eventually grow large enough to disturb the background flow. The instability continues to develop in this manner until the perturbation flow is of the same order of magnitude as the original background flow and is unable to extract any more energy from it. At the magnetopause, the speed of the magnetosheath flow increases with distance from the subsolar point [*Spreiter et*

al., 1966] thus increasing the energy available to the instability and suggesting that the instability, once started, will continue to develop for some distance tailward.

The KH instability in an initially laminar shear layer grows at an exponential rate through a linear phase of small perturbations and then enters a nonlinear regime dominated by large vortices. Once in the nonlinear phase, the develop-

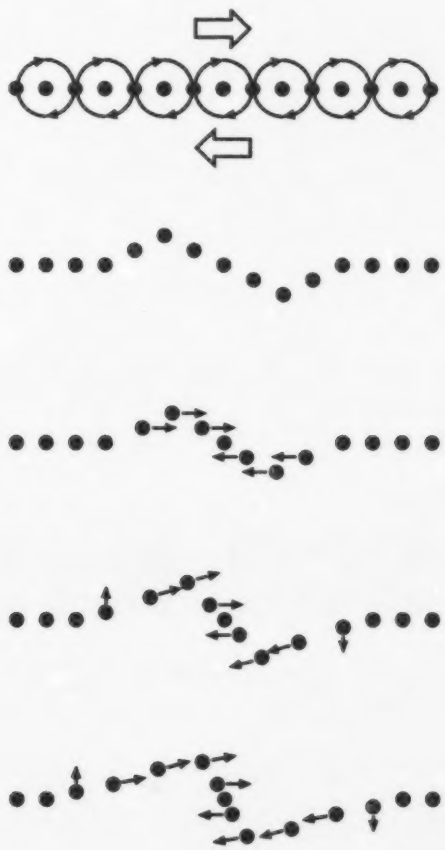


Figure 3. The development of the KH instability in a thin shear layer [after *Batchelor*, 1967]. The shear layer is equivalent to a line of vortices which, when perturbed, are brought together by the ambient flow field and result in a localized increase in the vorticity. That increase broadens the perturbation region and further increases the vorticity enhancement leading to an exponentially growing perturbation.

ment of the instability slows as it runs out of free energy and turns instead to multiple vortex mergings which make the boundary layer increasingly turbulent. Throughout the development of the instability, the boundary layer continually broadens under the influence of the turbulent diffusion [e.g., *McComb*, 1990] associated with the instability.

There are a number of ways in which a velocity shear layer may be stabilized. In conventional hydrodynamics, the presence of viscosity may broaden the shear layer quickly enough to make it stable to all wavelengths of perturbations [e.g., *Drazin and Reid*, 1981] as will the absence of an inflection point in the velocity profile [*Rayleigh*, 1880]. In MHD flows, the presence of a magnetic field introduces another means of stabilizing the shear layer. When the flow is perpendicular to the field, the instability behaves exactly as it does in the hydrodynamic case [e.g., *Miura and Pritchett*, 1982] and is not stabilized by the field. However, a flow-aligned component of the field can stabilize the shear layer against perturbations by making the energy required to bend the field lines greater than that available from the flow.

The absence of viscosity in magnetospheric plasmas and the velocity profile of the shear layer seen during satellite passes together ensure that the LLBL will be unstable to the KH instability in those regions where the magnetic field is perpendicular to the flow. Observations have shown [*Eastman et al.*, 1976] that the field on the earthward side of the boundary layer is dipole-like and directed northward in the equatorial plane of the magnetosphere but may undergo a rotation at the magnetopause which leaves it pointing in any direction roughly tangential to the magnetopause in the magnetosheath. Additionally, the LLBL field becomes increasingly tilted away from the northward direction with increasing geomagnetic latitude. Together these two effects increase the stability of the LLBL against the KH instability and have the potential to completely stabilize large regions of the boundary layer through energy losses to magnetic distortions.

The magnetic field lines that thread the LLBL extend to the Earth's ionosphere and loosely couple the two regions together. The vortical motions associated with the KH instability in the LLBL propagate along those field lines to the ionosphere where they are damped and reflected, now having the opposite sense of rotation, back to the LLBL [*Lysak*, 1990]. When they arrive at the boundary layer, they should act to reduce the vortical motions that generated them and thus damp the instability. However, the travel time required for the vortical motions to propagate to and from the ionosphere, >300 seconds [*Samson et al.*, 1992], is easily enough time for the KH instability to develop fully into its nonlinear stage before the initial vortical perturbation launched by the instability returns. Thus, ionospheric effects are irrelevant to the initial development of the KH instability in the LLBL.

There are other processes besides diffusive broadening and the KH instability that have been proposed to explain the existence of the LLBL. The impulsive penetration mechanism provides a relatively straightforward means of passing magnetosheath plasma into the magnetosphere but does not operate in quiet conditions when the incident solar wind flow is relatively slow and carries few significant density irregularities. Like impulsive penetration, the flux transfer event [*Russell and Elphic*, 1979] is another intermittent process which this time proceeds by merging magnetosheath and magnetosphere field lines at high geomagnetic latitudes to produce a kink (e.g., field line "1" in Figure 2) which is then swept through the LLBL by relaxation of the "tension" in the magnetic field. *Song and Russell* [1992] have advanced a mechanism which also takes advantage of high-latitude merging, but requires almost simultaneous merging in both hemispheres to connect a magnetosheath flux tube to magnetospheric field lines. The merged magnetosheath flux tube sinks into the boundary layer and moves tailward through the action of the interchange instability [*Southwood and Kivelson*, 1987]. *Sibeck* [1992] explains the dynamics of the LLBL as the result of another intermittent process which calls upon high amplitude pressure pulses

($\delta p/p \approx 1$) convecting past the LLBL in the magnetosheath to deform the magnetopause and temporarily bring a region of magnetosheath plasma across the average position of the magnetopause. A satellite passing through the boundary layer thus observes magnetosheath plasma inside the expected position of the magnetopause while still in the magnetosheath.

OBJECTIVES

The primary objective of my work has been to simulate the spatial development of the LLBL using a realistic computer simulation started from realistic initial boundary layer configurations. This emphasis on realism comes from the paucity of observations of the LLBL. To date, all of the observations made of the boundary layer have come from satellite passes which sample a very small volume of the time-varying boundary layer and thus fail to provide enough information to allow the development of a reliable theory of the LLBL as a whole. A realistic computer simulation, on the other hand, would be capable of reproducing the overall behaviour of the LLBL, provided that the assumptions inherent in the simulation are reasonable and that the simulation is started from realistic initial conditions. Such a simulation, when tempered by satellite observations, would serve as an excellent basis upon which to study the LLBL.

Once I had developed and verified the operation of a realistic LLBL simulation, it became possible to ask specific questions about the formation and dynamics of the LLBL and to devise initial conditions and simulation diagnostics which would directly address the questions. In particular, I wanted to determine whether the simulated LLBL would broaden with downstream distance and, if so, for what conditions. Rapid downstream broadening of the boundary layer would indicate that magnetosheath-magnetosphere transport processes were active in the LLBL and would support the idea that the LLBL is a significant site of solar wind-magnetosphere coupling.

Finding that the boundary layer does broaden rapidly with downstream

distance then led me to examine the mechanisms responsible for the broadening. The KH instability quite obviously plays a large role in the early development of the boundary layer, but saturates while the boundary layer is still quite thin making it relatively unimportant downstream where the boundary is at its broadest. I wanted to determine what other mechanism were responsible for broadening the boundary layer and, in particular, why, for some initializations, the boundary layer stops broadening.

LINEAR STABILITY

The statements made in the Introduction about the stability of velocity shear layers are made possible by a linear stability analysis of the shear layer. The stability analysis starts with a shear layer which is in equilibrium, though not necessarily stable, and then subjects it to small perturbations of the form

$$f_0(y) + \delta f(y) \exp\left[-i(kx - \omega t)\right], \tag{1}$$

where f is a physical variable, x and y are the streamwise and cross-stream coordinates, respectively and k and ω are the complex wavenumber and complex angular frequency of the perturbation. Substituting (1) into the equations describing the fluid permits the equations to be linearized and makes a solution for the perturbation eigenmodes possible. *Miura and Pritchett* [1982] found a numerically-soluble eigenvalue equation associated with these eigenmodes for the case of a shear layer in an ideal MHD fluid with the Doppler-shifted frequency, $\omega + kv_0(y)$, as the eigenvalue. The shear layer is unstable when the Doppler-shifted frequency has a positive imaginary component.

Setting either k or ω to a real constant and then solving the eigenvalue equation for the other quantity makes it possible to examine both the temporally and spatially growing KH instabilities. When k (ω) is a real constant, (1) shows that the instability will experience no spatial (temporal) growth and only temporal (spatial) growth will be possible. In both cases, the most unstable eigenmode associated with a particular shear layer configuration may be identified and, since it develops faster than any other mode, it should quickly come to dominate the shear layer. To date, the stability of the LLBL has been examined almost exclusively by assuming that only a temporally growing mode is present [e.g., *Ong and Roderick*, 1972; *Walker*, 1981; *Miura and Pritchett*, 1982; *Pritchett and Coroniti*, 1984; *Miura*, 1984; *Miura*, 1987; *Belmont and Chanteur*, 1989; *Miura*, 1990; *Rajaram et al.*, 1991; *Fujimoto and Terasawa*,

1991; *Miura*, 1992], with two exceptions [*Wu*, 1986; *Wei et al.*, 1990], in spite of the fact that the LLBL, like most physical shear layers, has a point of origin, the subsolar point, and develops spatially with distance from it.

Assuming a temporally developing solution in a stability analysis of an infinitely thin, incompressible shear layer permits derivation of a growth rate for the KH instability [*Southwood*, 1979]:

$$\mathrm{Im}(\omega)^2 = \frac{\rho_1\rho_2}{(\rho_1+\rho_2)^2}\left(\mathbf{k}\cdot(\mathbf{v}_1-\mathbf{v}_2)\right)^2 - \frac{(\mathbf{k}\cdot\mathbf{B}_1)^2+(\mathbf{k}\cdot\mathbf{B}_2)^2}{\mu_0(\rho_1+\rho_2)}, \qquad (2)$$

where ρ is the mass density and $\mathbf{v}$ is the velocity of the plasma, $\mathbf{B}$ is the magnetic field and the subscripts "1" and "2" indicate magnetosphere and magnetosheath values, respectively. Equation (2) illustrates the relative importance of the directions of the velocity, magnetic field and wave vectors: the instability grows fastest when the wave vector of the perturbation is parallel, or antiparallel, to the difference in velocities and perpendicular to the magnetic field. When the magnetic field has a component directed parallel, or antiparallel, to the wave vector, the growth rate is reduced and the shear layer can be completely stabilized by the magnetic field "tension" represented by the $\mathbf{k}\cdot\mathbf{B}$ terms. When the magnetic field and velocity vectors are aligned everywhere in the flow, the shear layer is everywhere stable so long as $|c_A|_{min} > (v_{max} - v_{min})/2$, where c_A is the Alfvén speed [*Kent*, 1968].

Equation (2) indicates that the growth rate of the instability becomes arbitrarily large for small wavelength perturbations. In reality, however, any physical shear layer will have a finite thickness which reduces the growth rate for short wavelength perturbations and introduces a lower limit on the perturbation wavelength. A numerical stability analysis for a realistic LLBL configuration (Case B, Table 1; the details of this configuration will be discussed later in the thesis) using a shooting method [*Press et al.*, 1986] to solve *Miura and Pritchett*'s [1982] eigenvalue equation for the spatially growing modes gives the

growth rate dependence on wavenumber shown in Figure 4. Instead of having an arbitrarily large growth rate for large wavenumbers, the maximum growth rate for this particular configuration of the LLBL occurs when , where is the half-width of the shear layer.

Another useful result of the stability analysis is an expression for the phase velocity of the instability wave [*Axford*, 1962]:

$$\mathbf{v}_{ph} = \left(\mathrm{Re}(\omega)/\mathrm{Re}(k^2)\right)\mathbf{k} = \left(\rho_1\mathbf{v}_1 + \rho_2\mathbf{v}_2\right)/\left(\rho_1 + \rho_2\right). \qquad (3)$$

This expression indicates that the shear layer perturbation is stationary in the center of mass reference frame. Note that (3) comes from assuming an infinitely thin shear layer in an incompressible fluid. A more general stability analysis that includes compressibility and allows the shear layer to have a finite thickness gives phase velocities that can differ significantly from those given by (3).

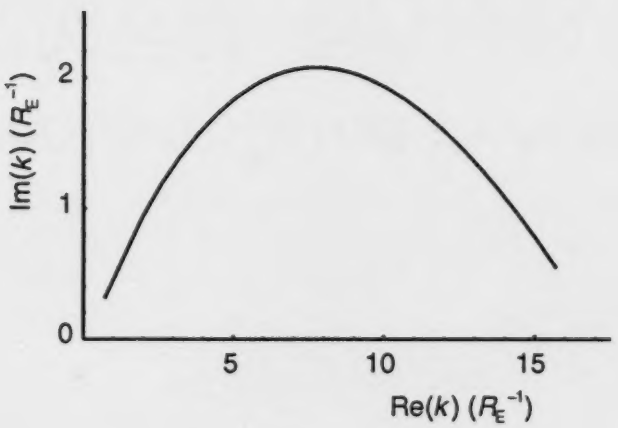


Figure 4. The spatial growth rate of the KH instability found by a linearized perturbation analysis for a realistic boundary layer configuration (Case B, Table 1; the details of this configuration will be discussed later in the thesis).

SIMULATION

The simulation is based on the ideal MHD equations:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{4}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) = -\nabla\left(p + B^2/2\mu_0\right) + (\mathbf{B} \cdot \nabla)\mathbf{B}/\mu_0, \tag{5}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{v}\mathbf{B}) = (\mathbf{B} \cdot \nabla)\mathbf{v}, \tag{6}$$

$$\frac{\partial p}{\partial t} + \nabla \cdot (p\mathbf{v}) = -(\gamma - 1)p\nabla \cdot \mathbf{v}, \tag{7}$$

where ρ, $\mathbf{v}$, $\mathbf{B}$, p and γ are the mass density, velocity, magnetic field, plasma pressure and adiabatic index ($\gamma = 5/3$ in this paper) of the plasma and μ_0 is the permeability of free space. Equations (4) and (7), the continuity equation and the adiabatic equation of state, take the same form as in conventional compressible fluid dynamics. Equation (5), the equation of motion, is the same as that of a conventional inviscid fluid with the addition of two terms which describe the effect of the magnetic field on the plasma. The first term in (5), $-\nabla\left(B^2/2\mu_0\right)$, is the pressure force associated with the magnetic field while the second term, $(\mathbf{B} \cdot \nabla)\mathbf{B}/\mu_0$, represents a "tension" force which attempts to straighten curved field lines [e.g., *Schmidt*, 1979]. Equation (6), Faraday's law, has no equivalent in conventional fluid dynamics. Writing the equations in the form above, with the continuity terms on the left-hand side, facilitates their implementation in the numerical finite difference scheme I use.

The relative simplicity of the ideal MHD equations makes them amenable to simulation but at the expense of their ability to fully describe the boundary layer plasma. Probably their most significant shortcoming is their inability to accu-

rately represent structures which have one or more of their dimensions on the order of one ion cyclotron radius (r_i) or less. Satellite passes of the magnetopause have shown it to have a minimum thickness of only three or four r_i [*Berchem and Russell*, 1982; *Sonnerup et al.*, 1990], a fact also borne out by particle simulations performed by *Berchem and Okuda* [1990] and *Gary and Sgro* [1990], thus putting the suitability of the ideal MHD equations for simulations of the boundary layer into question.

Although by definition no diffusion of any sort (i.e., density diffusion, viscosity, heat conduction or resistivity) is included in the ideal MHD equations, any grid-based algorithm that is used to solve the equations inevitably introduces some during the course of their solution. The finite difference scheme implemented in the simulation permits the amount of numerical diffusion to be accurately tuned so as to smooth only the sharpest features, those on the order of three numerical cell-widths in size or less, while leaving all larger scales untouched. The simulation takes advantage of this built-in filter by defining the numerical cell-width to be slightly greater than one r_i so that any structures on the simulation grid that begin to narrow so much that they are in danger of violating the large r_i condition are numerically diffused back to a relatively "safe" thickness of three r_i, much as they are in the particle simulations. Unlike physical diffusion processes, the algorithm effectively filters away only the highest spatial frequencies and so is able to preserve large features through many time steps without smoothing them appreciably. An ability to maintain large-scale structures is essential to simulations that attempt to follow forms that evolve slowly in comparison to wave transit times, such as the nonlinear KH instability.

My simulation is based on a modified version of the finite difference algorithm ETBFCT [*Boris*, 1976; *Book et al.*, 1981]. The modifications implement the improvements suggested by *Book and Fry* [1984] and *Zalesak* [1981] and place the algorithm in the multidimensional framework described by *Zalesak* [1979]. I further improved the algorithm by modifying it so as to reduce its tendency to

create "terraces" on steep gradients (see Appendix B). This modified algorithm is able to resolve sharp gradients while almost completely avoiding the effects of numerical dispersion and permits tuning to introduce an adjustable amount of short wavelength filtering while maintaining excellent amplitude and phase accuracy. *Grinstein and Guirguis* [1992] have shown that a much simpler form of this filter can emulate a physical viscosity thus suggesting that the new filter, with its flexibility, is at least capable of reproducing the effects of physical viscosity. Work under way by *Boris* [1990] may also confirm that the family of monotone methods, of which the modified algorithm is a member, actually are practical large eddy simulations [*Reynolds*, 1990] and as such, may be used to reliably simulate volumes that are so large that the smallest scales of motion must remain unresolved, as would be the case in future simulations of the entire LLBL or magnetosphere.

The simulation of the MHD equations that results from application of the

Figure 5. The density field some time after introducing a perturbation at the bottom center of the frame. The perturbation is applied to an otherwise uniform stationary plasma placed in an upward-directed magnetic field with its plasma parameters chosen so that $c_A : c_S : c_F = 3 : 4 : 5$. The black lines correspond to the analytically determined group velocity multiplied by the time at which the simulation snapshot was taken.

algorithm to (4), (5), (6) and (7) (see Appendix C) is capable of accurately representing the wave modes associated with the ideal MHD equations as well as the much slower dynamics seen in the development of the KH instability. Figure 5 compares the MHD wave field that develops from an initially localized density/pressure perturbation as simulated by the algorithm and the corresponding group velocity, determined analytically. The simulation results are free of numerically-generated dispersive ripples and agree well with the analytic results, though the slow magnetosonic mode wavefront, the inner "triangle" seen in Figure 5, appears under-resolved. Simulations of the KH instability also agree well with the linear stability theory during the initial linear phase of the instability. The growth rate predicted by the theory and found by simulation agree to within 2% (Figure 6) and the details of the nonlinear phase of the instability are well resolved, even on a low-resolution numerical grid.

I also investigated the sensitivity of the simulation to failures in the divergence-free state required of the magnetic field, $\nabla \cdot \mathbf{B} = 0$. Unlike *DeVore* [1991],

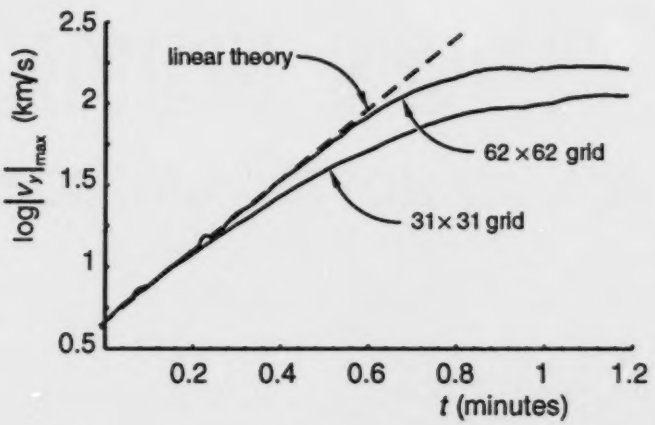


Figure 6. The temporal growth of the KH instability found by simulation (solid lines) and predicted by a linear perturbation analysis (dashed line) for a realistic boundary layer configuration (Case B, Table 1; the details of this configuration will be discussed later in the thesis). The slope of the lines gives the growth rate. $|v_y|_{max}$ is the maximum value of $|v_y|$ found on the simulation grid.

I do not treat Faraday's law, (6), any differently than the other ideal MHD equations and so must ensure that the inevitable numerical errors in **B** which make $\nabla \cdot \mathbf{B}$ nonzero do not introduce significant errors to any of the other variables. The test initialization consists of a stationary uniform plasma, with density and pressure set to the values used in the simulation, permeated by a monopolar magnetic field which has been defined so as to be as large as the largest $\nabla \cdot \mathbf{B}$ error observed in the simulation. The test showed that the monopole produced very small amplitude compressional waves but none of the field-aligned accelerations seen by *Brackbill and Barnes* [1980]. There are typically already many large amplitude compressional waves present by the time $\nabla \cdot \mathbf{B}$ errors become large enough to generate even small waves, so the effect of such numerically-generated waves may easily be discounted.

A simulation of the spatially developing KH instability requires inflow and outflow boundary conditions rather than the periodic ones used in simulating the temporally developing instability. Both boundaries should be nonreflecting so that waves incident on the boundaries leave the simulation grid without reflecting any energy back into the grid. In conventional fluid dynamic simulations, these boundary conditions are often satisfied simply by considering only those flows which are supersonic at both boundaries so that waves can neither propagate upstream to the inflow boundary nor reflect from the outflow boundary. In this case, the boundary conditions become quite simple: the inflow boundary has all quantities specified and the outflow boundary has Neumann boundary conditions with all boundary-normal gradients set to zero.

Unfortunately, the boundary conditions required of an LLBL simulation are complicated by the presence of submagnetosonic flows on the earthward side of the boundary layer which permit magnetosonic waves to propagate to and from the inflow and outflow boundaries. To accommodate the effect that these waves have on the inflow boundary, I implemented boundary conditions which would both respond to downstream conditions and ensure that the incoming plasma

has the desired plasma parameters. Much experimentation showed that inflow boundary conditions that specified values for ρ, $\rho\mathbf{v}$ and $\mathbf{B}$ but let p float as much as possible while being relaxed back to its fixed ambient value would allow the boundary to respond to magnetosonic waves without introducing secular variations into any variable [*Grinstein et al.*, 1991]. Specifically,

$$\rho_1 = \rho_{\text{in}}, \tag{8}$$

$$(\rho v_x)_1 = (\rho v_x)_{\text{in}}, \; (\rho v_y)_1 = 0, \tag{9}$$

$$\mathbf{B}_1 = \mathbf{B}_{\text{in}}, \tag{10}$$

$$p_1 = p_2 - \delta(p_2 - p_{\text{amb}}), \tag{11}$$

where the subscripts 1, 2 and "in" refer to a computational cell at the inflow boundary (In every other section of this paper, the subscripts 1 and 2 indicate magnetosphere and magnetosheath values, respectively), the cell immediately inward of it and a value fixed at the inflow boundary, respectively, p_{amb} is the ambient pressure, $\delta = \Delta t/\tau$ is the relaxation constant, Δt is the length of the time step, $\tau = l/c_{\text{max}}$ is the relaxation time, l is a length scale characteristic of one wavelength of the upstream propagating waves and c_{max} is the speed of the upstream waves at the inflow boundary. The coordinate system used in the simulation is shown in Figure 1. The boundary condition on p is important because it permits the development of cross-stream pressure gradients which are essential for self-excitation of the shear layer [*Grinstein et al.*, 1991]. I tried to apply similar boundary conditions to $\mathbf{B}$, but the fact that $\mathbf{B}$ is a vector quantity that must everywhere satisfy a vector relation, $\nabla \cdot \mathbf{B} = 0$, significantly complicates the boundary conditions, particularly when one has the possibility of a component of $\mathbf{B}$ normal to the inflow boundary as in the LLBL and magnetosheath.

The outflow boundary condition is rather different from that for the inflow. In addition to having to respond without reflections to magnetosonic waves, the outflow boundary must also permit the convecting vortices generated by the KH instability to exit the simulation without producing disruptions at the boundary. Boundary conditions that assume that all structures leaving the simulation space convect through the outflow boundary at the phase speed of the fastest growing KH instability mode satisfy this condition. This convection or wave boundary condition [*Grinstein et al.*, 1991; *Givoli*, 1991] operates by assuming that conditions at the boundary are well-represented by

$$\left(\frac{\partial}{\partial t} + v_{out}\frac{\partial}{\partial x}\right) f = 0, \tag{12}$$

where f represents ρ, $\rho\mathbf{v}$, $\mathbf{B}$ or p, v_{out} is the speed at which structures are carried off the grid. Discretizing (12) gives

$$f_I^{n+1} = f_I^n - \frac{v_{out}\Delta t}{\Delta x}\left(f_I^n - f_{I-1}^n\right), \tag{13}$$

where the superscripts indicate the time level and I denotes a computational cell at the outflow boundary. Values for V_{out} are determined by first identifying the most unstable spatially developing mode for a particular configuration of the LLBL and then determining the phase velocity of that mode. After being determined in (13), p_I^{n+1} is relaxed towards p_{amb} using a condition similar to (11) in order to better define what would otherwise be an under-specified boundary condition on p.

The inflow and outflow boundary conditions described above are far from ideal. Neither boundary condition can deal with transitory effects, such as waves, without generating spurious waves of their own. Ideal boundary conditions would permit waves and passively convected structures alike to be carried across simulation boundaries as if those boundaries were not present. Work in compressible hydrodynamics has come close to this ideal [*Thompson*, 1990] but

no similar set of general boundary condition for MHD fluids has yet been developed. Probably the most dangerous characteristic of the present boundary conditions is that they reflect some of the outgoing wave energy back into the simulation space. Hydrodynamic simulations have shown that numerical outflow boundary conditions are sometimes responsible for reinitiating the KH instability through boundary-generated waves [*Reynolds*, 1990] and that physically-generated waves produced during the nonlinear stage of the instability can also reinitiate the instability [*Grinstein et al.*, 1991] thus indicating a need for caution in dealing with waves of any origin in the vicinity of the boundaries.

The problem of boundary-generated waves may be dealt with by introducing enough wave energy at the inflow boundary to overwhelm the upstream travelling waves [*Reynolds*, 1990]. Adding a random perturbation to the pressure at every cell on the magnetosheath side of the inflow boundary creates both non-propagating "pseudosound" fluctuations [*Matthaeus et al.*, 1991] and waves, some of which impinge on the velocity shear layer where they introduce perturbations which help initiate the instability. The disturbed conditions in the simulated magnetosheath that result from these perturbations have a strong basis in reality: the shocked plasma constituting the magnetosheath does contain large amounts of wave energy [e.g., *Engebretson et al.*, 1991] and will likely act as a trigger for the development of the KH instability in the LLBL.

The remaining boundary conditions are straightforward. The boundaries parallel to the magnetopause are perfectly conducting walls which act to reflect all waves incident on them back into the simulation space. Work done in hydrodynamics has shown that the distance between a wall and the shear layer has little effect on the development of the shear layer for subsonic flows [*Chen and Morrison*, 1991], but can significantly alter the development of a supersonic shear layer by coupling together different regions of the shear layer through wall-reflected shock waves [*Zhuang et al.*, 1990]. The flows in the simulations presented here are submagnetosonic so the sole criterion used in choosing the

wall positions is that they should be far enough away from the shear layer that vortices generated by the KH instability do not touch them. The final two boundaries, parallel to the equatorial plane and forming the floor and ceiling of the simulation space, do not exist in this simulation because it is confined to two dimensions.

SIMULATION BOUNDARY LAYER

Any simulation of the boundary layer must start from plausible initial conditions. The observed boundary layer is the time-dependent result of complicated interactions between the magnetosheath and magnetosphere plasmas and is not well enough understood to permit construction of an initialization that is both realistic and physically self-consistent. Instead of trying to develop such an initialization, I model the initial LLBL as a simple boundary layer between two regions, the magnetosheath and magnetosphere, each consisting of a uniform plasma with plasma properties based on actual satellite observations. Such a configuration is simple enough to be made physically self-consistent and ensures that the simulation starts from an initialization that is well-understood.

The magnetosheath and magnetosphere parameters used in the simulations are based on those observed during an ISEE-1 satellite pass through the boundary layer from 0130 to 0230 UT on 17 August 1978 [*Eastman et al.*, 1985] and shown in Figure 7. This pass is particularly well-suited to simulation because the boundary layer at the time of the pass was broad and well-defined with a single, sharp magnetopause crossing and an easily identified LLBL and remained in approximate fluid and magnetic pressure balance for the duration of the pass: having each pressure balance separately across the boundary layer eliminates the possibility of the interchange instability [*Southwood and Kivelson*, 1987] developing and adding another element of complexity to an already sufficiently complex situation. The crossing occurred in the dusk sector at 1636 LT, putting it relatively far upstream in the LLBL and thus increasing the chance

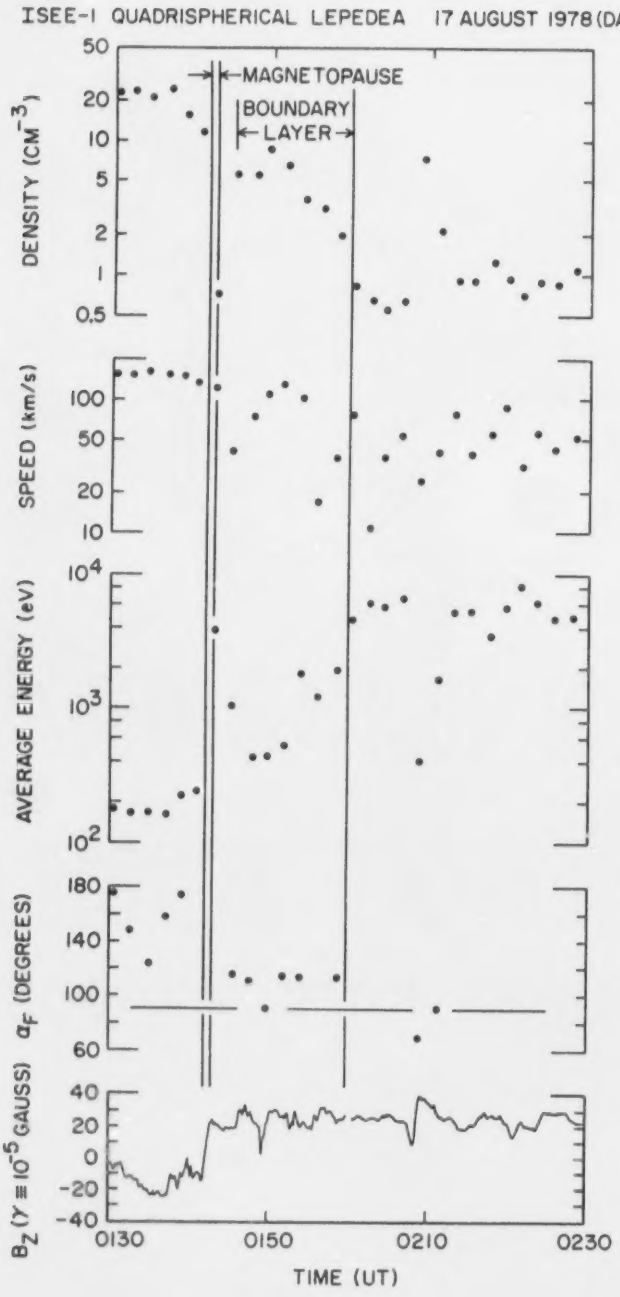


Figure 7. The plasma and magnetic field parameters observed during the inbound ISEE-1 magnetopause crossing of 17 August 1978 [from *Eastman et al.*, 1985]. α_F is the angle between the bulk ion flow vector and magnetic field vector and is equivalent to the parameter ϕ used in this thesis.

that the boundary was still relatively undeveloped. After slightly adjusting the magnetosheath and magnetosphere values seen during the satellite pass in order to put the two regions into exact pressure balance, I obtain the values given in the left two columns of Table 1. Some new parameters are introduced in the table: n is the plasma number density, ϕ is the angle between $\mathbf{B}$ and $\mathbf{v}$ (both vectors are assumed to be tangent to the magnetopause), a_{LLBL} and a_{MP} are the half-widths of the LLBL and magnetopause, respectively, $y_{\mathrm{MP}} - y_{\mathrm{LLBL}}$ is the displacement of the magnetopause and LLBL centers, and M_{C} and M_{F} are the convective and magnetosonic Mach numbers, respectively. Here, and in the remainder of this thesis (except where indicated otherwise) the magnetospheric parameters are identified with the subscript "1" and the magnetosheath values with the subscript "2".

The parameters defining the simulated magnetosheath and magnetosphere are given in the right two columns of Table 1 and are a compromise between observational reality and computational necessity. The simulation parameters are chosen to retain as much of the character of the observed parameters as possible while increasing the efficiency of the simulation. A linear stability analysis of the boundary layer based on the simulation values of the plasma parameters shows that the fastest growing KH instability mode takes less than half the time to develop as the same boundary layer configuration based on the observed values. Using the observed values to define the boundary layer would thus require more than twice as much computer time to reach a particular stage in the evolution of the instability as using the simulated values would and yet both sets of parameters result in boundary layers with approximately the same spatial growth rate ($\mathrm{Im}(k_{\mathrm{max}})$ in Table 1). Ensuring that the convective Mach number, M_{C}, of the boundary layer remains near its observed value further increases the similarity between the two parameter sets [*Sandham and Reynolds*, 1989b]. Having both the spatial growth rate and the convective Mach number associated with the values used in the simulation set as closely as

	observed [a]		used in simulation (case B) [b]	
	magneto-sheath	magneto-sphere	magneto-sheath	magneto-sphere
n (cm^{-3})	40	0.8	2	1
v (km/s)	150	0	449	53
p (nPa)	0.77	0.77	0.39	0.39
B (nT)	23	23	16.4	16.4
ϕ (°)	140	90	120	90
r_i (km)	49	344	217	307
a_{LLBL} (km)	?		382	
a_{MP} (km)	?		382	
$y_{MP} - y_{LLBL}$ (km)	?		573	
$\mathrm{Im}(k_{max})$ (R_E^{-1})	2.46 [c]		2.68 [c]	
v_{ph} (km/s)	104 [c]		225 [c]	
M_C ($=(v_2 - v_{ph})/c_{F_1}$)	0.288 [c]		0.337 [c]	
M_F ($=(v_2 - v_1)/c_F$)	0.94	0.13	0.78	0.55
c_S (km/s)	139	981	439	621
c_A (km/s)	80	567	253	358
c_F (km/s)	160	1133	507	717
β ($=2\mu_0 p/B^2$)	3.6	3.6	3.6	3.6

[a] Parameters of the observed LLBL seen in Figure 7.

[b] Parameters of the simulated LLBL.

[c] Calculated using the simulation values for ϕ_2, a_{LLBL}, a_{MP} and $y_{MP} - y_{LLBL}$ and after changing reference frames to make $v_1 = 0$.

Table 1. Parameters of the LLBL crossing.

possible to those determined from the observations should make the simulation results directly applicable to the LLBL.

In addition to specifying the magnetosphere and magnetosheath parameters, the initialization must also describe the initial form of the transition between them, the LLBL itself. The initial LLBL transition is specified by the functions

$$\rho_0(y) = \rho_1 + f_{\mathrm{LLBL}}(y)(\rho_2 - \rho_1),$$

$$v_0(y) = v_1 + f_{\mathrm{LLBL}}(y)(v_2 - v_1), \tag{14}$$

$$\phi_0(y) = \phi_1 + f_{\mathrm{MP}}(y)(\phi_2 - \phi_1),$$

where the subscript 0 indicates that the values are initialization values, $\mathbf{v}_0 = v_0\hat{\mathbf{x}}$ and the profile functions are defined as

$$f_{\mathrm{LLBL}}(y) = 1 + 1/2 \tanh\left[(y - y_{\mathrm{LLBL}})/a_{\mathrm{LLBL}}\right],$$

$$f_{\mathrm{MP}}(y) = 1 + 1/2 \tanh\left[(y - y_{\mathrm{MP}})/a_{\mathrm{MP}}\right]. \tag{15}$$

The initial half-widths are set to make both the magnetopause and LLBL three r_i thick and the displacement between the magnetopause and LLBL centers, $y_{\mathrm{MP}} - y_{\mathrm{LLBL}}$, is set to 2.2 r_i. The centers are displaced to satisfy observations (e.g., Figure 7) that indicate that the magnetopause is most often seen at the edge of the boundary layer and to create conditions which make it easier for the instability to develop [*Ogilvie and Fitzenreiter*, 1989; *Rajaram et al.*, 1991].

The parameters given in Table 1, along with (14) and (15), constitute the base configuration of the LLBL modelled in this paper, referred to as case B in the remainder of this thesis. There are two other cases considered in what follows, both of which have one aspect of their initializations changed but are otherwise identical to case B: case A has no rotation of the magnetic field at the magnetopause and case C has coincident shear and current layers. The three cases are

ordered according to the growth rate of the KH instability for each case, as given in Table 2. The stabilizing effect of the magnetic field is responsible for reducing the growth rates in cases B and C. A stability analysis has shown that for the worst case of a magnetosheath magnetic field that is completely flow-aligned, only the case C boundary layer configuration is completely stabilized against the KH instability.

		Case		
		A	B	C
		$\phi_1 = \phi_2 = 90°$ $y_{MP} - y_{LLBL}$ = 573 km	$\phi_1 = 90°$, $\phi_2 = 120°$ $y_{MP} - y_{LLBL}$ = 573 km	$\phi_1 = 90°$, $\phi_2 = 120°$ $y_{MP} = y_{LLBL}$
$\mathrm{Re}(2\pi/k_{max})$	(R_E)	0.847	0.797	0.928
$\mathrm{Im}(k_{max})$	(R_E^{-1})	2.16	2.08	1.65
$\mathrm{Re}(\omega_{max}/2\pi)$	(mHz)	53.3	54.6	46.4
simple [a] v_{ph}	(km/s)	317	317	317
full [b] v_{ph}	(km/s)	288	277	274

[a] $\mathbf{v}_{ph} = (\rho_1 \mathbf{v}_1 + \rho_2 \mathbf{v}_2)/(\rho_1 + \rho_2)$

[b] from full linear stability analysis

Table 2. Parameters and phase velocities of the KH instability for the three boundary layer configurations.

NUMERICAL RESULTS AND DISCUSSION

CASE B

To set the stage for the long, spatially developing LLBL simulations that follow, I first present the results of a small simulation of the LLBL which uses periodic boundary conditions at the inflow and outflow boundaries. The analytical and computational simplicity afforded by the choice of periodic boundary conditions has made them a popular choice for many previous studies of the LLBL, as indicated in the Linear Stability section. Periodic boundary conditions permit an analytical solution for the temporally developing perturbation modes of the KH instability in the LLBL which may then be used to initialize a simulation. Finding the fastest growing perturbation eigenmode possible for the base LLBL configuration (case B) described in the previous section and initializing the simulation with it ensures that the instability develops from a pure mode. Experiments with different grid sizes have shown that a square 31×31 grid (with two additional boundary cells in each direction) composed of square $0.025\,R_E \times 0.025\,R_E$ cells resolves the eigenmode well enough to allow the instability develop at the analytically determined rate without much diffusion.

Figure 6 compares the temporal growth rate found analytically for the fastest growing mode with the growth rate found by simulation. The points plotted are the maximum values of $|v_y|$ on the grid for each time step; the slopes of the curves give the growth rate. The simulation growth rate agrees very well with the analytic one early in the linear phase of the instability but decreases prematurely as gradients steepen and become susceptible to the numerical filter (described in detail in Appendix B). Doubling the resolution of the simulation almost completely eliminates the effect of the filter during the linear phase by raising the threshold of the filter well above the spatial frequency associated with the instability but requires eight times as much computer time and would

require more memory for a simulation of the spatially evolving KH instability than is currently available. However, a slightly conservative growth rate for a simulation of the temporally developing KH instability helps to increase confidence in the results of later, more demanding simulations of the corresponding spatially developing instability by ensuring that the structures that develop during the course of such simulations are significant enough to have survived the effects of the numerical filter.

In spite of the low resolution and slightly reduced growth rate of the small, temporally developing simulation of the KH instability, the instability develops into the nonlinear phase without suffering from any of the more obvious numerical errors (e.g., see Appendix B). Figure 8 shows contour plots of the density, the z-directed component of the current and vector field plots of the planar components of the velocity and magnetic field after one minute. The vortical velocity field is well-established and has begun to wind the density and magnetic field up into spirals. The magnetic field is not as wound up as the density because the displacement of the magnetopause and LLBL centers means that a vortex developing in the LLBL at the center of the shear layer must grow for some time before it is large enough to begin to draw in the magnetosheath magnetic field. The current layer (magnetopause) develops a paired current sheet where the magnetic field is most distorted by the vortex and over the course of the simulation, it more than doubles in strength at the leading edge of the vortex where the magnetosheath flow compresses the field against the vortex.

Extending the simulation grid and changing the periodic boundary conditions to inflow and outflow boundary conditions permits the simulation of long, spatially evolving KH instabilities. The simulation grid is now 255×77 cells (with two additional boundary cells in each direction), but has the same grid resolution and is based on the same LLBL configuration (case B) as the small, temporally developing simulation just described. Instead of attempting to find an analytical form for the most unstable spatially developing shear layer, I apply

the noisy inflow boundary conditions described earlier. Figure 9 shows the compressional wave field one minute into the simulation, before the KH instability has had enough time to develop appreciably beyond the noise level. Recall that the perturbation is applied to the density and pressure values at the inflow boundary on the magnetosheath side and generates short wavelength waves

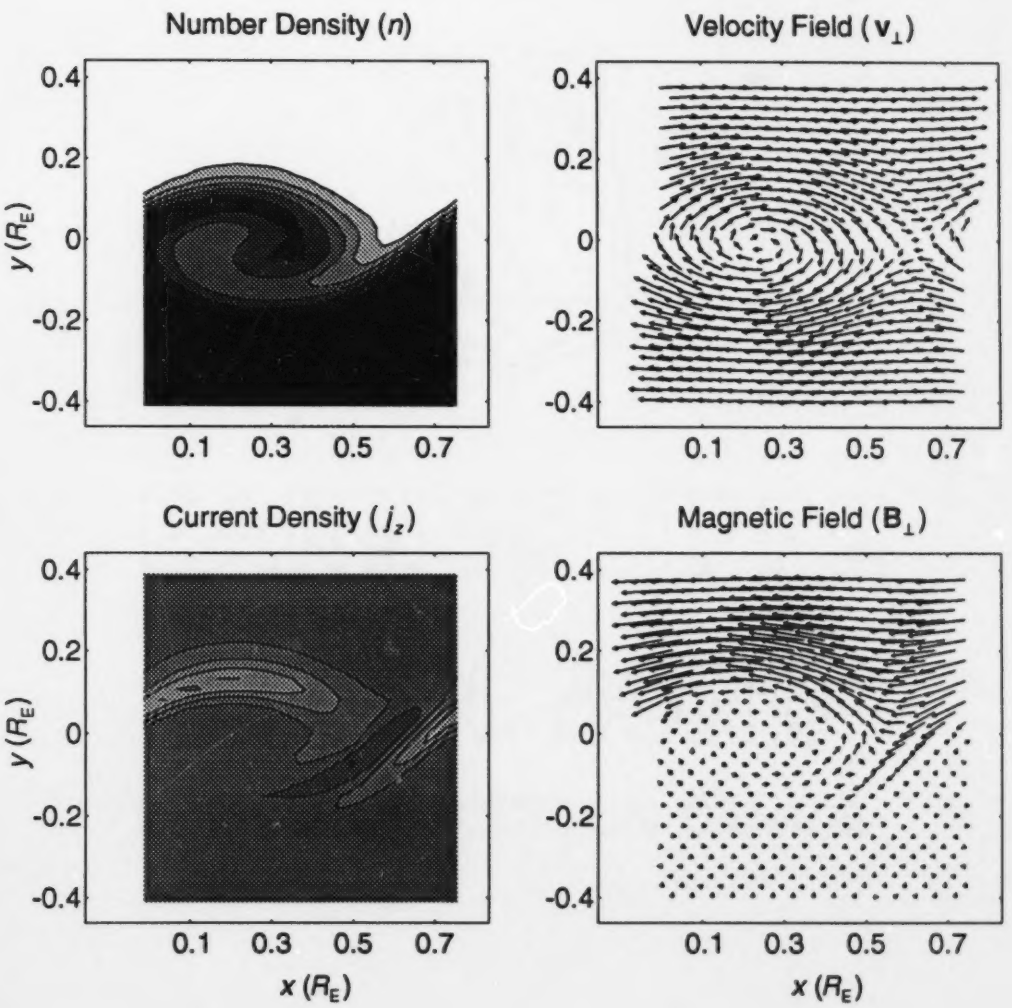


Figure 8. A snapshot taken at $t = 1.0$ minute of a temporally developing shear layer started from the boundary layer configuration of case B. The contours of density are at intervals of 0.18 cm^{-3} starting at 1.06 cm^{-3}, the contours of z-directed current density are at intervals of 4.8×10^{-9} A/m^2 starting at -7.2×10^{-9} A/m^2.

which are subsequently smoothed by the numerical filter.

Figure 10 shows the spatial growth of the KH instability for three LLBL configurations. Quantitatively demonstrating the spatial growth of the instability is more difficult than showing temporal growth in a temporally developing shear layer (see Figure 6) because of the difficulty in consistently identifying an exponentially developing feature in a flow which both grows and oscillates in space rather than one that just grows in time as the temporally developing instability does. The points plotted in Figure 10 correspond to the extrema of the oscillating and growing transverse velocity, v_y, extracted from sixteen consecutive snapshot files spanning the period from 2 to 5 minutes. The extrema are identified by first finding the absolute maximum of $|v_y|$ for a fixed x, but varying y-coordinate, and then finding all of the local maxima in the resulting series of maxima:

$$|v_y|_{max} = \mathrm{lmax}\left[\underset{x'=x}{\mathrm{amax}}\left[|v_y(x', y)|\right]\right], \tag{16}$$

where lmax[ ] finds local maxima and $\underset{x'=x}{\mathrm{amax}}$[ ] finds the absolute maximum for a fixed x.

Case B of Figure 10 (the other cases will be discussed in the following section)

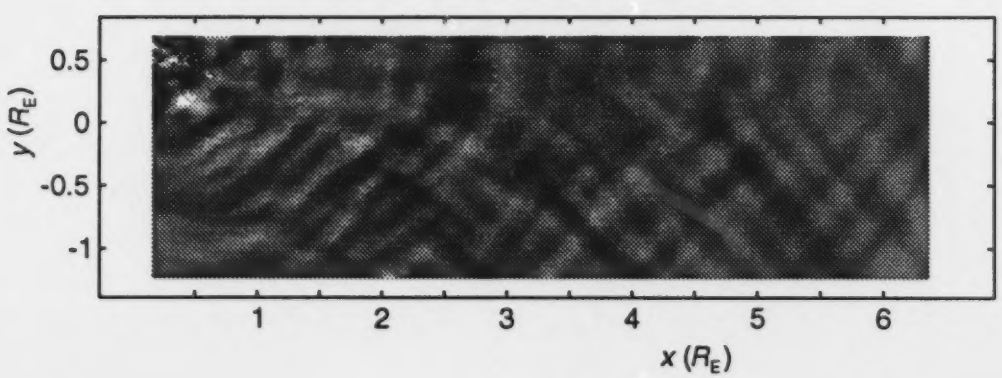


Figure 9. The compression wave field ($\nabla \cdot \mathbf{v}$) generated by the noisy magnetosheath inflow boundary.

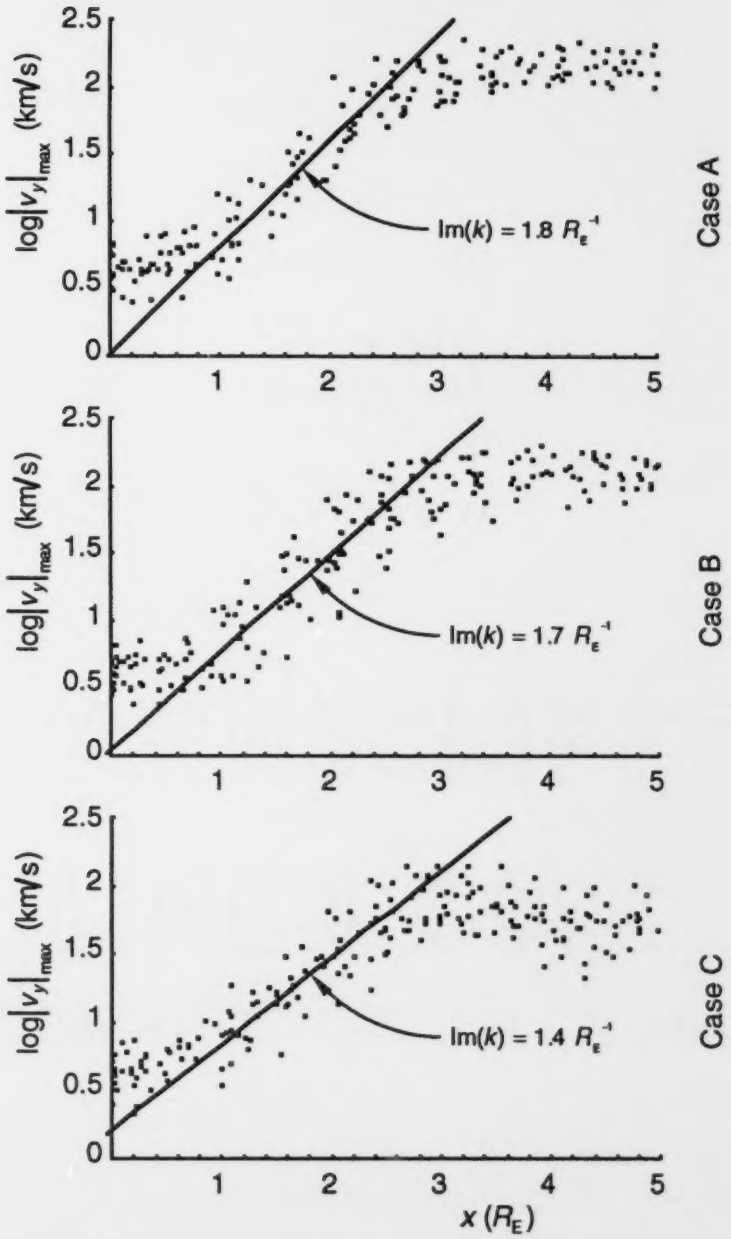


Figure 10. The spatial growth of the KH instability for three boundary layer configurations. The lines are the best fits to $|v_y|_{max}$ in the regions of exponential growth; their slopes give growth rates. For a given time step, $|v_y|_{max}$ is the set of local maxima found from a series made up of the largest $|v_y|$ for each downstream x position. The plot shows all of the $|v_y|_{max}$ for $t = 2.0,\ 2.2,...,5.0$ minutes.

shows that the shear layer experiences none of the rapid growth associated with the KH instability until 1 R_E downstream of the inflow boundary. It is likely that the instability is actually growing in this region but is obscured by the noisy inflow conditions. Beginning at 1 R_E downstream of the inflow boundary, the perturbation velocity increases exponentially through two orders of magnitude over a 1.5 R_E long region to saturate at ~30% of the total velocity jump across the shear layer. The solid line shown in the figure is the line that best fits the region of exponential growth. The spatial growth rate given by the slope of this line is 20% lower than the maximum growth rate found analytically for the most unstable mode, as indicated by comparison with Figure 4. The difference is likely due to the numerical filtering already seen at work in Figure 6.

Note that the growth of the velocity perturbation associated with the development of the KH instability is limited to a rather small region only 1.5 R_E long. Beyond that region, the maximum perturbation velocity remains relatively steady but continues to drive nonlinear processes which are responsible for the continued spatial development of the simulated LLBL. In the actual LLBL, the total velocity jump across the layer increases with distance tailward as the magnetosheath flow accelerates to solar wind speeds suggesting that the region of growth for the velocity perturbation will be much more extended and will continue to develop even as nonlinear effects become important.

Figure 11 shows snapshots of the state of the simulation taken at $t = 3.4$ minutes for the three LLBL configurations. Comparison of the spatially developing boundary layer that develops for the case B configuration with that seen in the corresponding simulation of the temporally developing LLBL (Figure 8) shows that although the small-scale (vortex) structure is quite similar, the overall structures of the boundary layers are very different. For small differences in velocity across the shear layer, the two scenarios are related by a Galilean transformation [*Kaul*, 1988] which associates time in the temporal layer with space in the spatial layer through the transformation

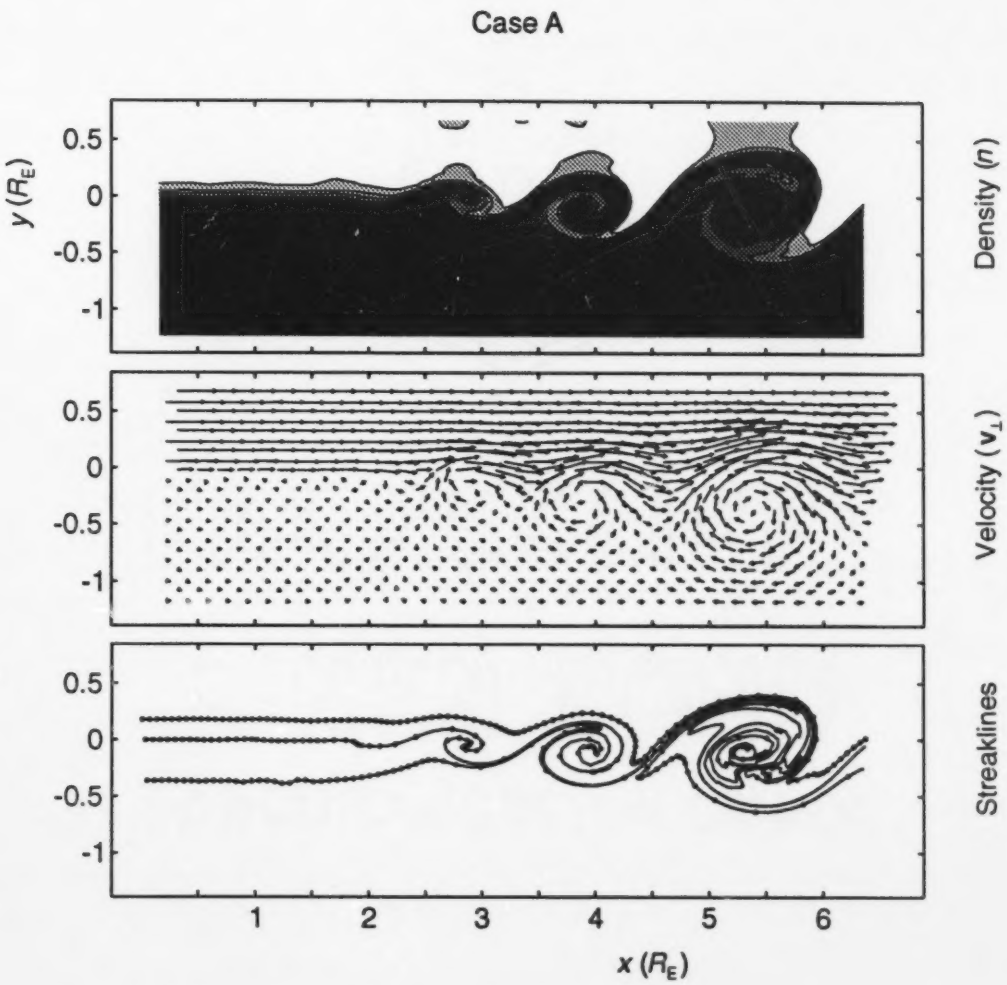
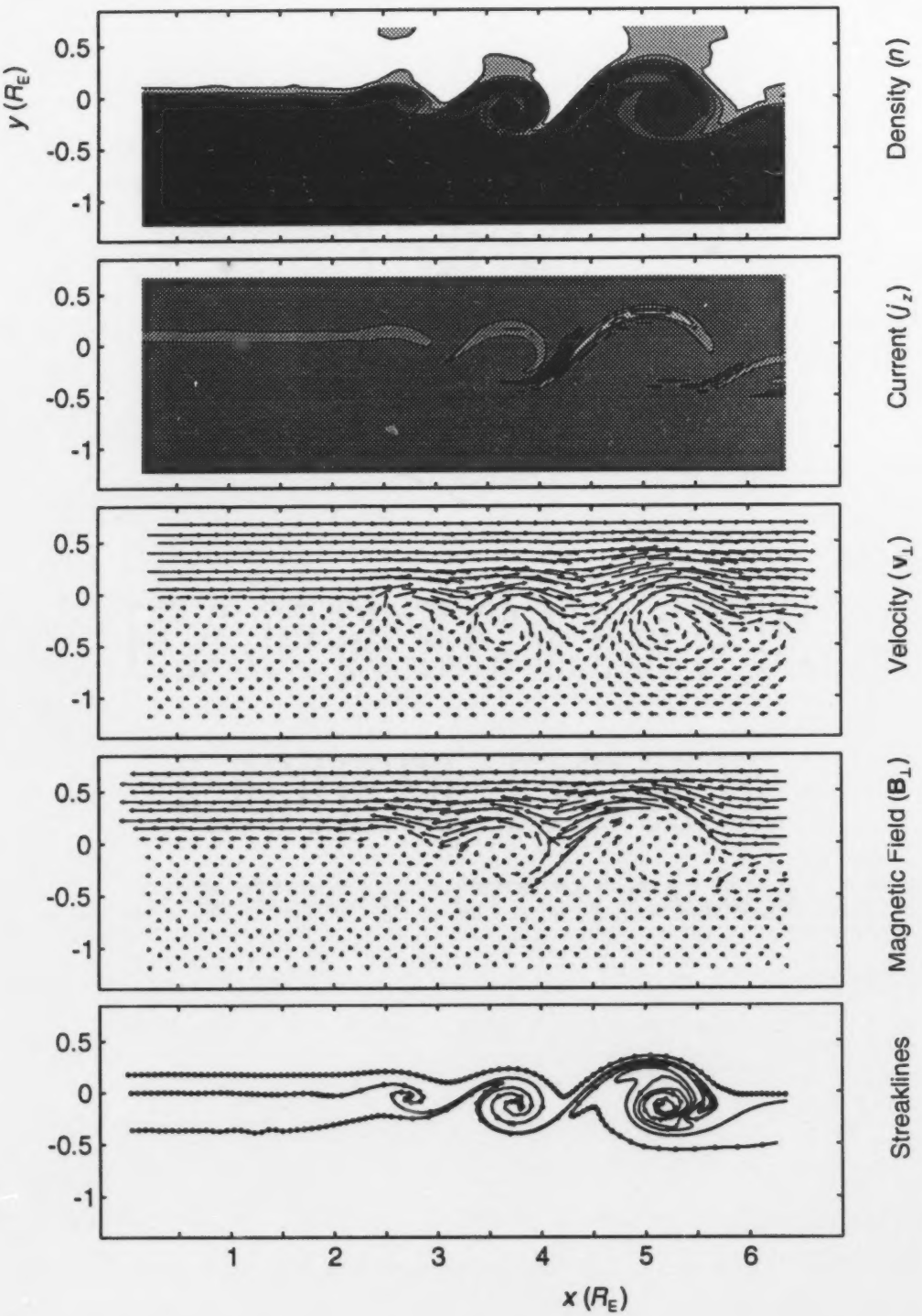
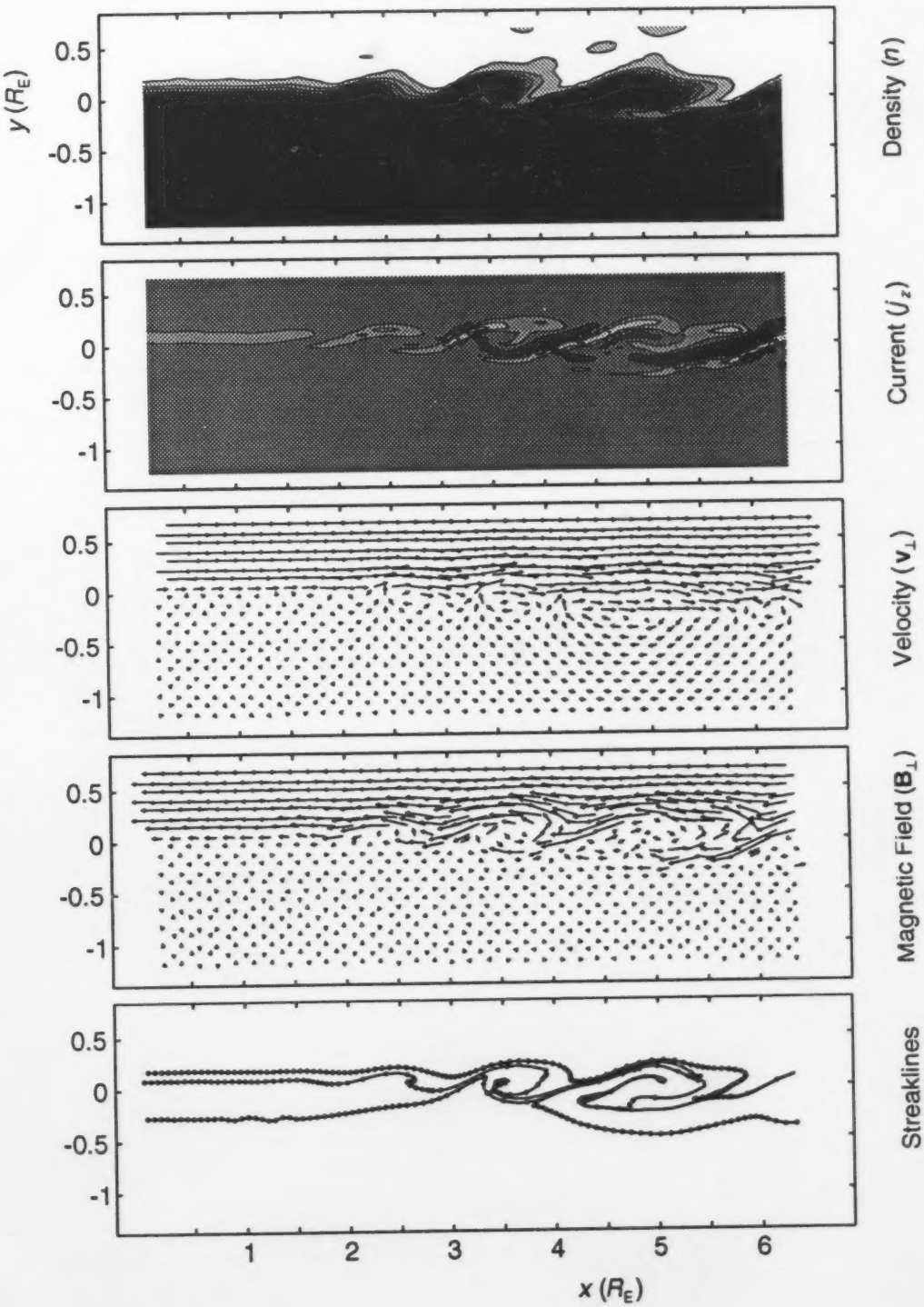


Figure 11. Snapshots of the three simulated LLBL configurations taken at $t = 3.4$ minutes. The contours of density are at intervals of 0.18 cm^{-3} starting at 1.06 cm^{-3} and the contours of z-directed current density are at intervals of 8.9×10^{-9} A/m^2, centered about the zero level. The current density and magnetic field vector plots are missing for case A because the magnetic field is everywhere directed normal to the plane of the simulation space.

Case B


y (R_E)
0.5
0
-0.5
-1
Density (n)
Current (j_z)
Velocity (v_⊥)
Magnetic Field (B_⊥)
Streaklines
1
2
3
4
5
6
x (R_E)

Case C


y (R_E)
0.5
0
-0.5
-1
Density (n)
Current (J_z)
Velocity (v_⊥)
Magnetic Field (B_⊥)
Streaklines
1
2
3
4
5
6
x (R_E)

$$x = v_{ph} t', \tag{17}$$

where x is the downstream coordinate in the spatial layer, t' is time in the temporal layer and v_{ph} is the phase velocity of the KH instability waves. Although this transformation is rather inaccurate for all of the boundary layer configurations considered in this thesis because of the large difference in velocity across the shear layer, it shows how events that occur early (late) in the temporal layer map to upstream (downstream) regions of the spatial layer and indicates that information that is carried upstream in the spatial layer would be equivalent to information carried back in time in a temporal layer. For example, an event, such as a vortex merging, occurring in a temporally developing shear layer is certainly incapable of affecting the prior development of the layer but when such an event occurs in a spatially developing shear layer, it can induce an upstream change in the shear layer which subsequently alters the downstream flow and the development of the instability [*Grinstein et al.*, 1991].

The snapshot presented in case B of Figure 11 is taken shortly after the instability assumes a quasi-stationary state in which the overall form of the flow no longer changes, although the details of the turbulence do. The turbulent mixing consists of vortices which are generated, grow, sometimes merge and then propagate off the simulation grid. The first two vortices, immediately downstream of the inflow boundary as seen in the density plot, are similar in form to the single vortex produced by the periodic simulation because they have developed from the shear layer in much the same way as that vortex. The third vortex appears somewhat confused in structure because it is the result of a merging between two vortices and is in the process of merging with a third. Vortex merging occurs as a result of other, slowly developing subharmonic modes of the KH instability which grow to become large, but weak, vortices capable of rotating a pair of smaller vortices into close proximity so that they merge [e.g., *Corcos and Sherman*, 1984]. Vortex merging creates large, persistent vortices which appear able to survive far downtail in the magnetosphere [*Sanderson et*

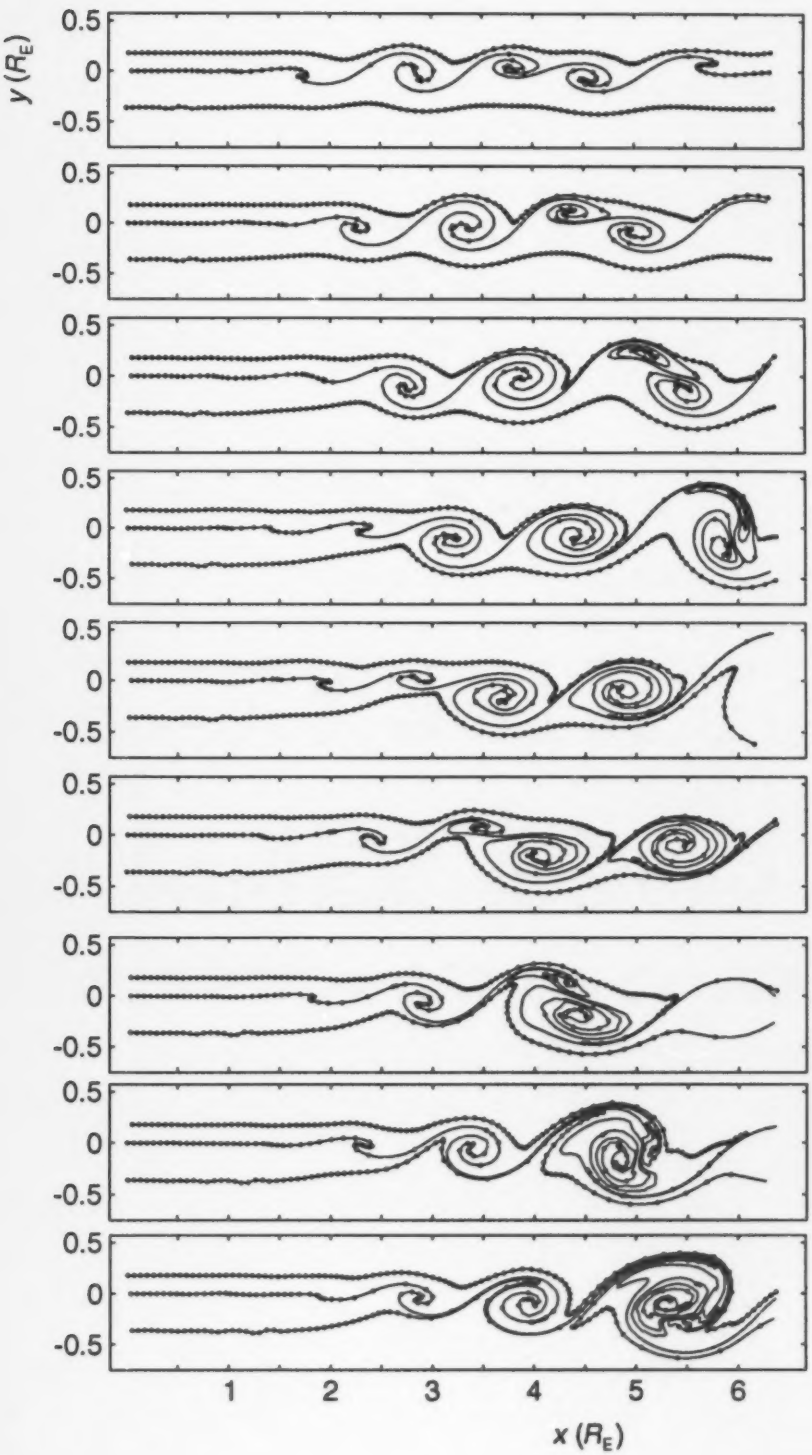
al., 1986].

Figure 12 shows a sequences of streakline snapshots leading up to the snapshots presented in Figure 11. These plots show the lines traced out by massless particles released into the flow field at the inflow boundary. The particles are placed in the magnetosphere, magnetosheath and the center of the shear layer at the inflow boundary and are left to move in response to the continually evolving flow field. The first snapshot in the series is early in the development of the instability before the entire shear layer has felt the effect of the inflow perturbations. At the inflow end of the streakline positioned at the center of the shear layer is a small kink that eventually develops into the third vortex shown in Figure 11. As it develops, the vortex draws in material along the shear-centered streakline leading to a concentration of the tracer particles at the center of the vortex and an absence in the regions between it and neighboring vortices, much as described by *Rosenhead* [1931], shown in Figure 3 and seen in experimental fluid dynamics [e.g., *Panides and Chevray*, 1990]. Fluid continuity requires that the plasma drawn into the vortex be replaced by plasma that comes from somewhere else. That plasma comes from outside the shear layer and its entry into the shear layer is responsible for distorting the magnetosheath and magnetosphere streaklines and giving them their varicose appearance. The regions in between the vortices develop into stagnation regions where flows converge (seen clearly in the velocity field of Figure 8) and are always found between the varicose pinches. As the instability develops further, the magnetosphere streakline becomes fully entrained into the shear layer and the tracer particles on it are accelerated downstream, as indicated by the increased spacing between particles and seen by satellite [*Mitchell et al.*, 1987]. Note, however, that the magnetosheath streakline remains relatively unaffected by the development of the instability and is not entrained into the boundary layer in spite of its proximity to the shear layer.

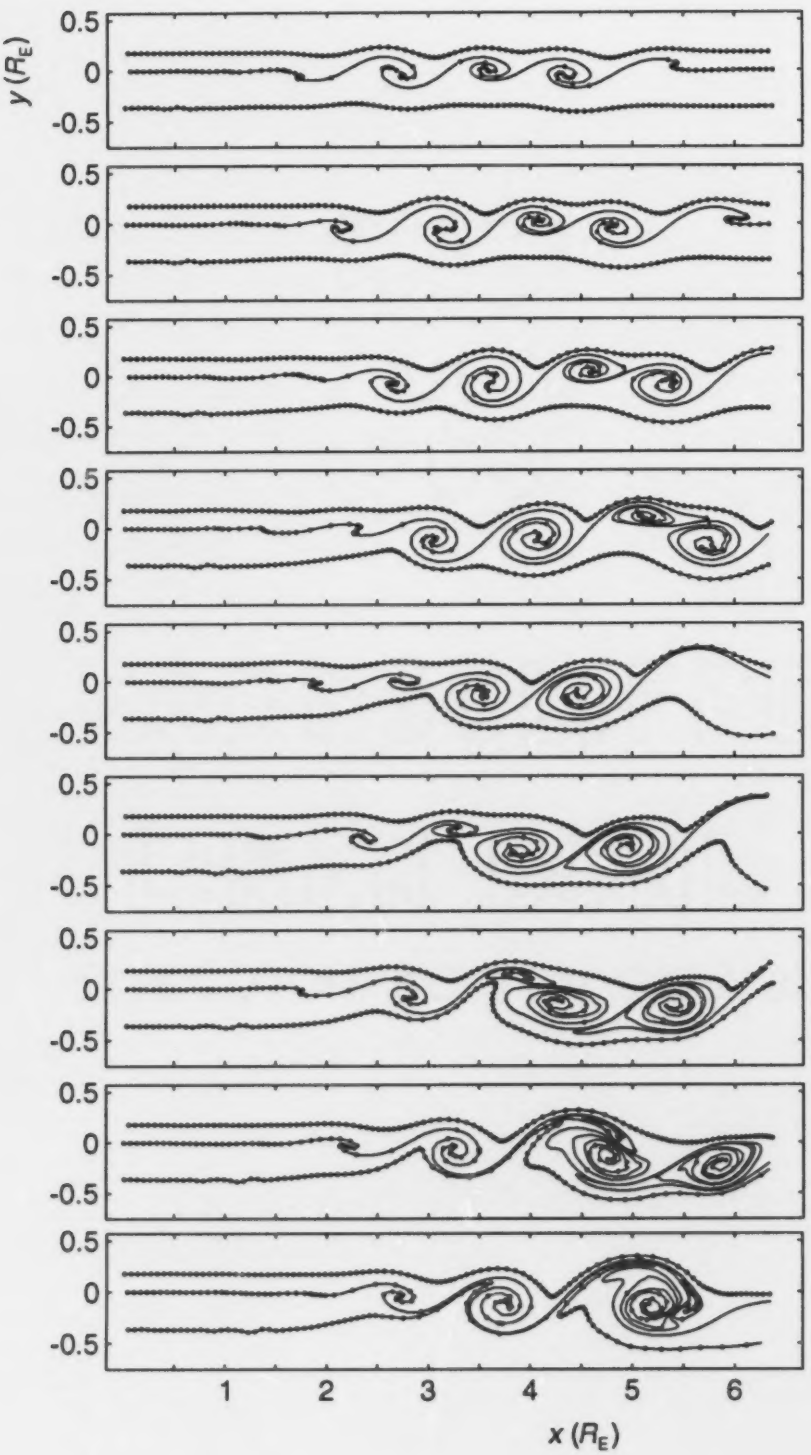
The streakline plots demonstrate how the roll-up and subsequent merging of

Figure 12. Snapshots of streaklines for the three simulated LLBL configurations. The upper streakline is in the magnetosheath, the middle one is at the center of the shear layer and the lower one is in the magnetosphere. The snapshots are taken at intervals of 12 seconds from $t = 1.8$ to 3.4 minutes.

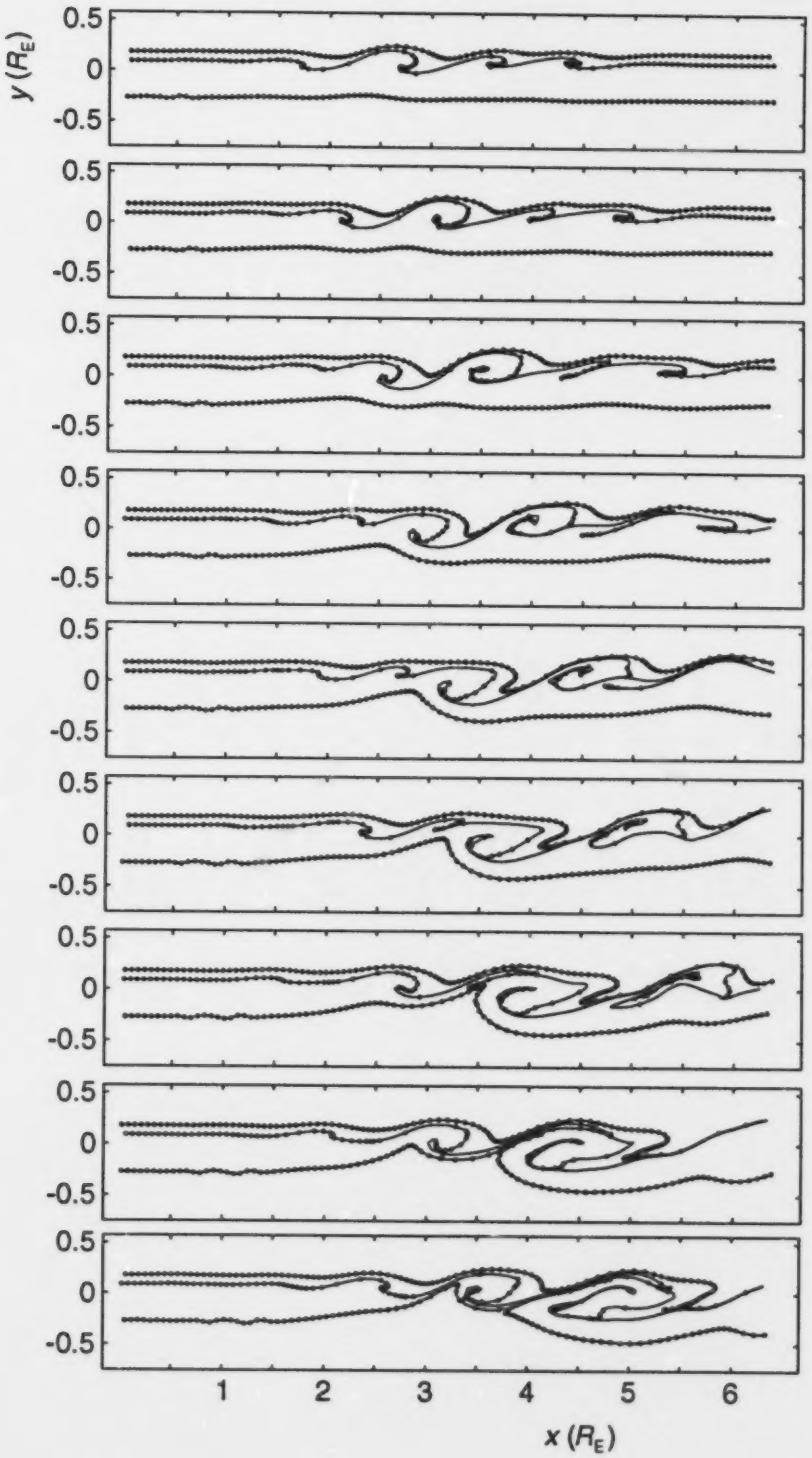
Case A



Case B



Case C



vortices in the presence of diffusion can thoroughly mix the magnetosheath and magnetosphere plasmas to form a boundary layer of roughly intermediate properties. The vortex roll-up draws the dense plasma that has diffused across the magnetopause from the magnetosheath and the lighter magnetosphere plasma on the other side of the shear layer into close proximity at the two stagnation regions that flanking a vortex. The boundary layer between the two plasmas in the vicinity of the stagnation region is very thin in comparison to the thickness of the magnetopause and is subject to diffusive processes such as those usually invoked to account for the magnetopause thickness (see Figure 13). The thin sub-boundary layer between the two plasmas is maintained well into the vortex as it rolls up but may eventually succumb to diffusion and lead to a plasma of intermediate properties at the center of the vortex where the diffusive process has had the most time to work. Homogenization also occurs when vortices merge: fine structure inside the vortices is lost to diffusion as the vortices are flattened, one against the other, in a stretching and folding process characteristic of chaotic flows [e.g., *Ottino*, 1990]. Through these processes, the magnetosheath and magnetosphere plasma may be mixed to eventually become indistinguishable

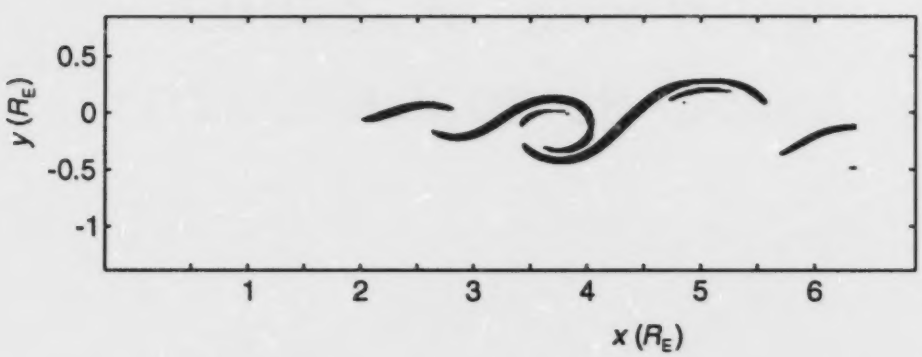


Figure 13. $|\nabla\rho|$ at t = 3.4 minutes for boundary layer configuration of case B. The contour level is chosen to identify regions where $|\nabla\rho|$ has increased beyond its value at the inflow boundary and thus indicates regions which are susceptible to diffusion.

from one another in the boundary layer. The density plot for case B in Figure 11 gives an indication of the extent to which this mixing has progressed in the third vortex. The plasma in it is homogenizing under the combined effects of the mixing process and the numerical filter built into the simulation.

CASES A, B AND C

Including the other two cases in the analysis now permits the discussion to be expanded to examine the effect that the differences between the three cases have on the development of the LLBL. Recall that cases A and C are variants of case B (described in Table 1): case A has no rotation of the magnetic field at the magnetopause and case C has coincident shear and current layers. The simulations of all three cases are subjected to exactly the same noisy inflow boundary condition through the same choice of initializing seed for the random number generator.

Comparing the snapshots for the three cases (Figure 11) shows that the different boundary layer configurations respond with striking similarity to the noisy inflow boundary condition: all three sets of snapshots show three vortical structures, the first two of which have never merged and the third of which has merged once and is in the process of merging again. This is in spite of the fact that the boundary layers simulated in the three cases are different enough from one another to have rather different growth rates and wavelengths for the fastest growing mode and indicates the importance of perturbations in determining the form of the boundary layer. Work in hydrodynamics has demonstrated the importance of the inflow perturbation in controlling the spatial-development of the shear layer [*Sandham and Reynolds*, 1989a]; the same is certain to be true for MHD and the LLBL.

The absence of a flow-aligned component of the magnetic field in case A removes the inhibiting effect that such a magnetic field has on the development of the instability. The vortices in case A are larger than those in case B because

they grow without having to bend magnetosheath field lines, an action which takes energy from the vortex and puts it into the magnetic field. The boundary layer that develops in case C, for which the shear layer and magnetopause are coincident, is strongly modified by energy transfer from the flow to the magnetic field. Initially, the developing instability easily overcomes the "tension" of the magnetic field lines and vortices develops, though slower than in cases A and B. Eventually, however, the increasingly energy-intensive deformations of the magnetic field require more energy than is available in the flow and the vortex roll-up stops. The second and third vortices seen in the density and streakline plots for case C appear to be cases of arrested vortex development. The second vortex has only recently stopped developing while the third vortex has stopped and been partially detached from the body of magnetosphere plasma. This detached vortex is connected to the magnetosphere by a thin filament of plasma which continues to stretch in later timesteps.

Moffatt [1978] has described the effect of differential rotation, such as that seen in any vortex generated by the KH instability, on an initially uniform magnetic field which is perpendicular to the axis of rotation. He assumes an incompressible infinitely conducting fluid with an azimuthal, axisymmetric flow field to find a generated component of the field,

$$\mathbf{B}_{gen} = B_0 rt\Omega'(r)\cos(\theta - \Omega(r)t)\hat{\boldsymbol{\theta}}, \tag{18}$$

where B_0 is the initially uniform field strength, ρ is the distance from the axis of symmetry, $\Omega(r) = v(r)/r$, $v(r)$ is the flow field and θ is the azimuthal coordinate and $\hat{\boldsymbol{\theta}}$ is its direction. The exact amount of energy put into the magnetic field in this way, U_B, depends on the details of the flow field, but assuming a localized rotational motion and integrating the energy over all space finds that

$$U_B \propto (B_0 v_0 t)^2, \tag{19}$$

where v_0 is a velocity characteristic of the rotation. Thus, even for an exponen-

tially increasing velocity field, such as that associated with developing KH vortices, the rate at which energy is deposited into the magnetic field will eventually increase faster than the rate of increase of flow energy and the vortex roll-up will stop. In the simulations, the flow fields and the kinetic energy associated with the vortices are seen to reach a steady-state when there is no flow-aligned component of the field, as for case A of Figure 10. In the presence of a flow-aligned component of the field, such as case C, the flow kinetic energy is lost to magnetic field deformations as the vortices wind up and the shear layer is thus stabilized.

The stabilizing effect of the magnetic field in case B is intermediate between its effect in cases A and C. The shear layer develops vortices which are initially free to grow without being impeded by the magnetic field but then slow their development as they impinge on the magnetosheath (compare the size of the third vortex seen in cases A and B). The simulation grid used is too short to allow the development of a large vortex to be followed long enough to see whether it stops developing once it entrains enough magnetosheath field, as they do in case C, but the previous argument suggests that they will.

The presence of a flow-aligned magnetic field in the velocity shear layer, as in case C, also inhibits the entrainment of plasma from outside the shear layer. The magnetosphere streakline is entrained at about the same rate for both cases A and B, but much more slowly for case C. The magnetosphere streakline seen in Figure 11 bounds a very thin filament of magnetosphere plasma which contributes little to the composition of the boundary. Even at a later time during a brief interval without vortex merging (see Figure 14), the entrainment flows of cases A and B are able to deliver large amounts of plasma from the magnetosphere into the boundary layer vortices while there is very little vortex-induced mixing across the shear layer for case C.

In all three cases, the magnetosheath streakline remains mostly unentrained for the length of the simulation grid. Case A does show the instability having

some success in entraining the streakline while case B hardly entrains the streakline at all because the flow-aligned component of the magnetic field resists deformation by the entrainment velocity field. Even case C, with its shear layer coincident with the magnetopause, only experiences marginal success before losing too much energy to magnetic field deformations.

Averaging the results for each case over a one or two minute interval starting at $t = 2.6$ minutes illustrates the persistent features of the simulated LLBL

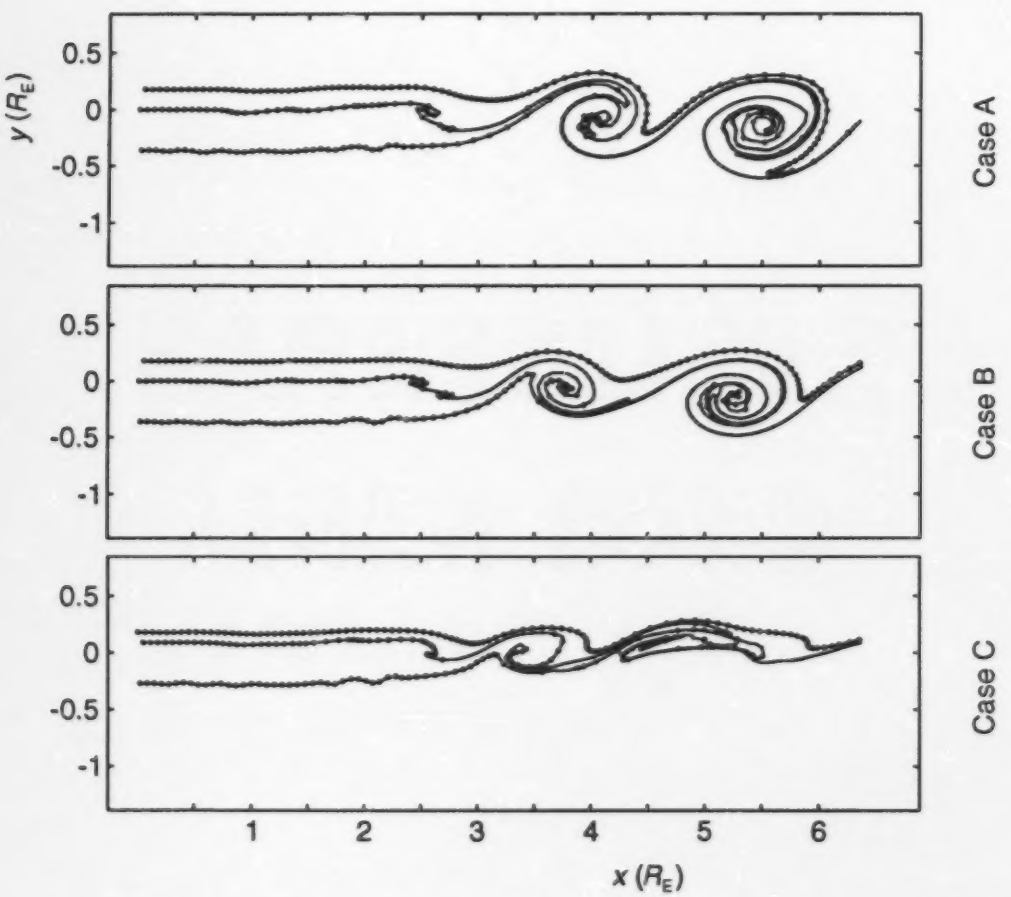


Figure 14. Snapshots of streaklines for the three simulated LLBL configurations. The upper streakline is in the magnetosheath, the middle one is at the center of the shear layer and the lower one is in the magnetosphere. The snapshots are taken at $t = 5$ minutes.

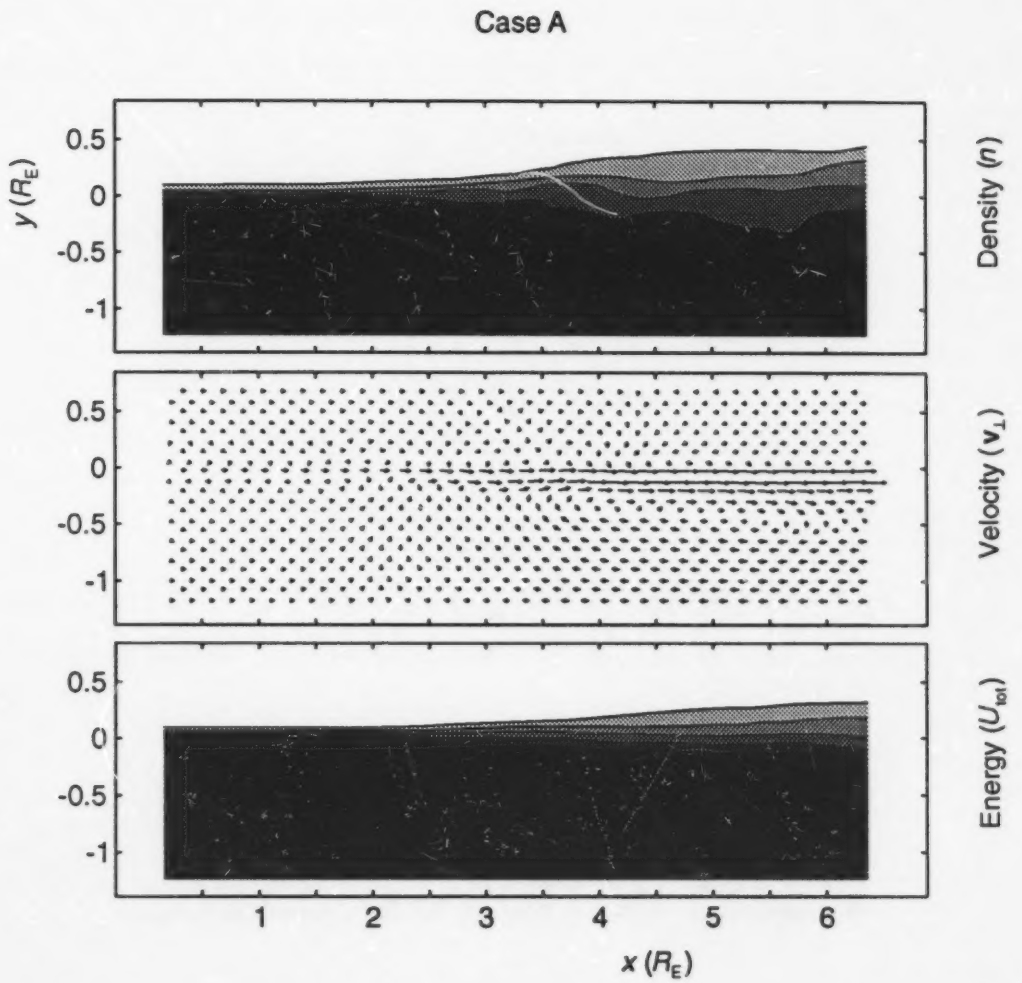
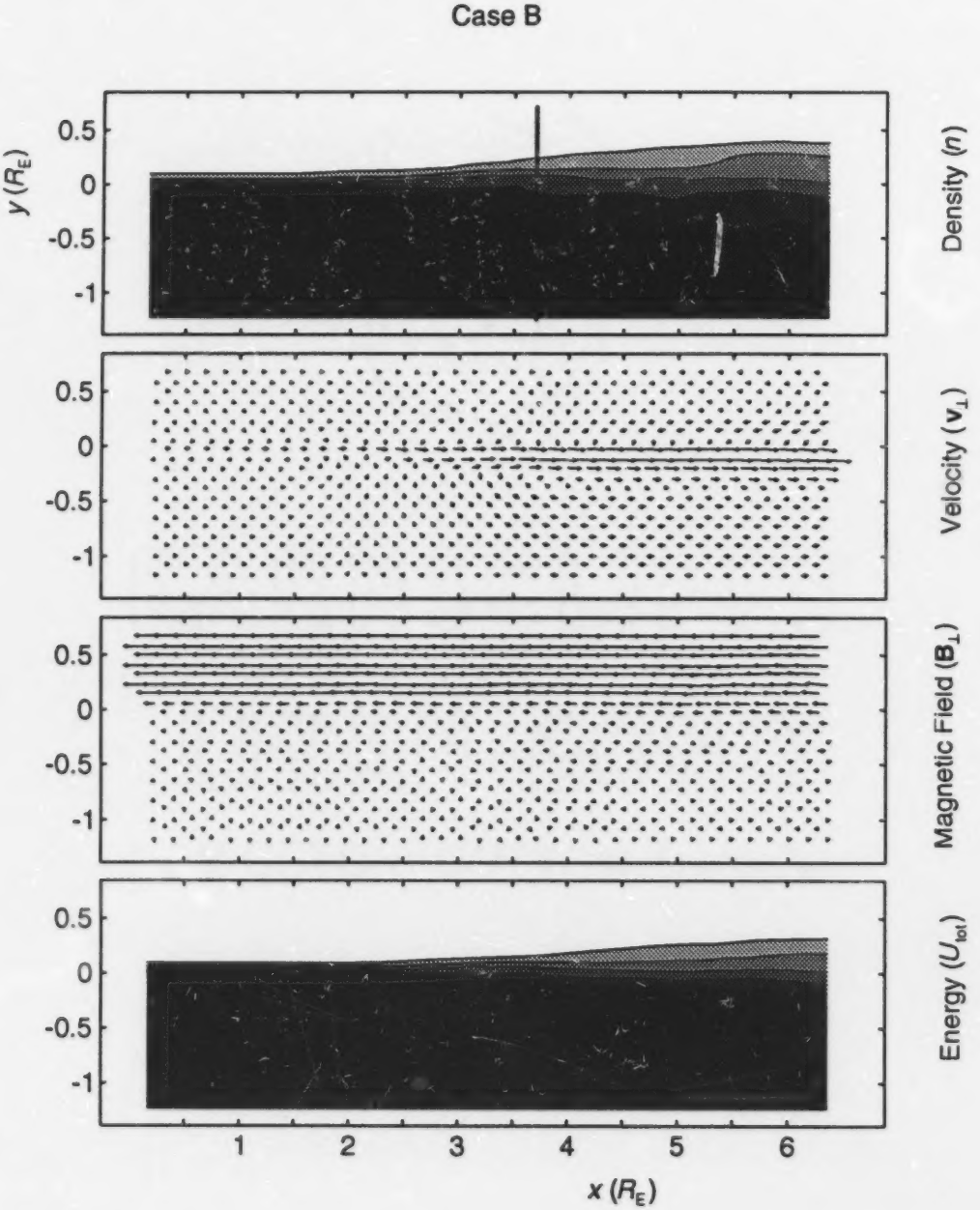
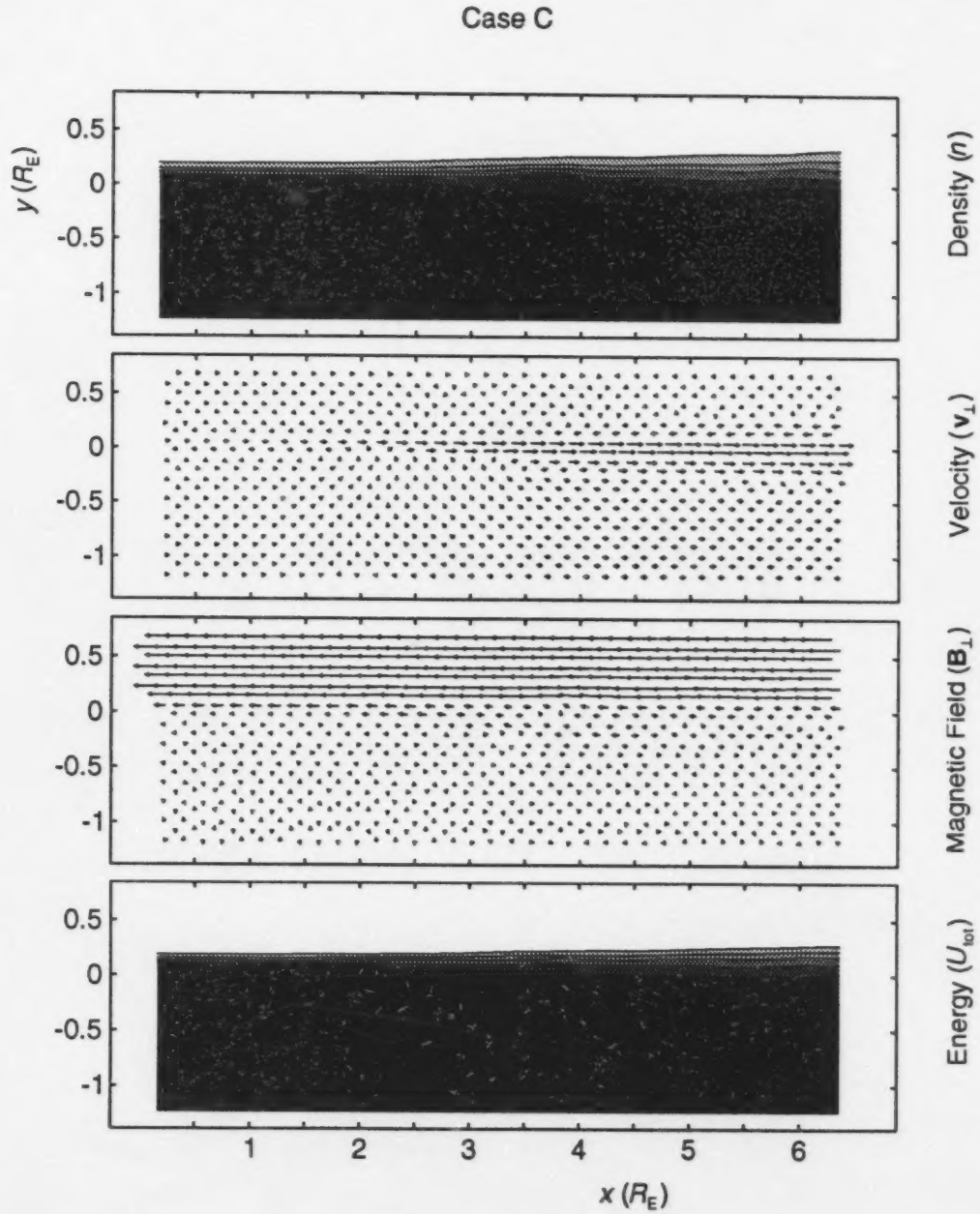


Figure 15. Time-averaged boundary layer for the three simulated LLBL configurations. The velocity field shown is the average field minus the initial field. Case A is averaged from $t = 2.6$ to 3.825 minutes, case B is averaged from $t = 2.6$ to 3.75 minutes and case C is averaged from $t = 2.6$ to 4.45 minutes. The contours of density are at intervals of 0.18 cm^{-3} starting at 1.06 cm^{-3} and the contours of energy density are at intervals of 53×10^{-12} J/m^3 starting at 690×10^{-12} J/m^3. The magnetic field vector plot is missing for case A because the magnetic field is everywhere directed normal to the plane of the simulation space.


Case B
y (R_E)
0.5
0
-0.5
-1
Density (n)
Velocity (v_⊥)
Magnetic Field (B_⊥)
Energy (U_tot)
1
2
3
4
5
6
x (R_E)


Case C
y (R_E)
0.5
0
-0.5
-1
Density (n)
Velocity (v_⊥)
Magnetic Field (B_⊥)
Energy (U_tot)
1
2
3
4
5
6
x (R_E)

(Figure 15). Ideally, the averaging interval, T_{avg}, will be long enough to eliminate all transient effects, but a long average is not practical for time-consuming simulations such as these. Instead, I make use of the quasi-stationarity of the shear flow and adjust T_{avg} to try and make the state of the boundary seen at the end of the period match that seen at the beginning thereby minimizing the spurious contributions made to the average by the starting and ending conditions. This method approximates the continually evolving boundary layer by one that goes through the same phase of development once every T_{avg} minutes and allows a more reliable definition of the "appropriate period of time" used by *Lu and Wu* [1991] in averaging the results of a hydrodynamical supersonic shear layer simulation.

Time-averaging the simulated boundary layer makes a comparison possible between it and satellite observations of the boundary layer which are also time-averaged. The relative absence of small-scale structure in the average simulated boundary layer (see Figure 16), in comparison to the observed boundary layer (see Figure 7), is not merely a result of the averaging: Figure 16 and case B of Figure 15 come from a 69 second average during the course of the simulation while each point plotted in Figure 7 is the result of a 128 second average of data taken from instruments during the satellite pass [*Eastman et al.*, 1985] thus suggesting that the simulation average should show more structure than the observed boundary. The fact that the observed boundary layer contains small-scale structure even after undergoing a long average confirms that the boundary layer is not developing under the almost steady conditions assumed in the simulation, but instead is significantly modified by variations in the surrounding plasma which are likely due to variations in the solar wind [*Sibeck*, 1992].

The large-scale features of the time-averaged plasma density, velocity and magnetic fields seen in Figures 15 and 16 are similar to those seen during the satellite pass shown in Figure 7. The magnetopause, as identified by the magnetic field rotation for cases B and C, remains narrow in comparison to the

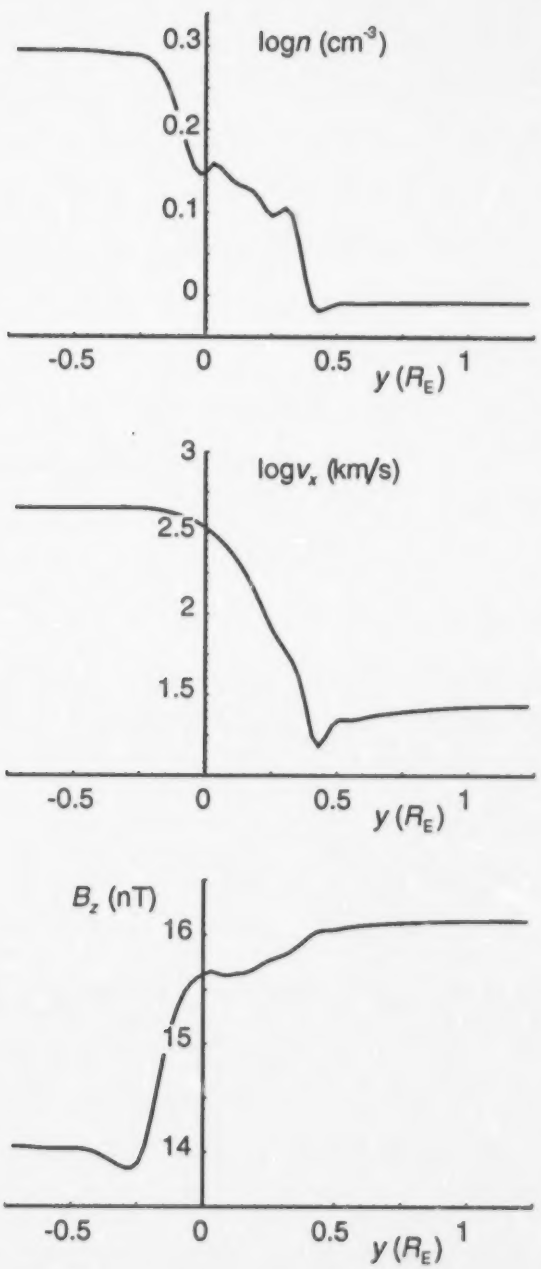


Figure 16. Cross-section of the average shear layer along the line plotted in Figure 15 for case B. The logs of n and v_x are plotted and the coordinate axis reversed to aid comparison with Figure 7. At $t = 0$, the center of the velocity shear layer is at $y = 0$.

LLBL in spite of the very large perturbations made to it by the KH instability and stays well outside of the shear layer, partially justifying the assumption of an initial displacement between the centers of the velocity shear and current layers which was made earlier in determining which LLBL configurations to study. The transition from magnetosheath to magnetosphere density levels starts slightly on the magnetosheath side of the average magnetopause and continues well into the earthward side, showing a plateau with intermediate density in the middle of the boundary, like that observed by *Paschmann et al.* [1978], thus supporting *Sckopke et al.*'s [1981] proposal that eddies produced by the KH instability are responsible for mixing the boundary layer plasma. The velocity boundary layer extends slightly across the average magnetopause into the magnetosheath, but most of it is situated on the earthward side of the magnetopause.

The boundary layers produced by each of the three cases presented in Figure 15 are once again rather similar despite the differences between the cases. In each case, the boundary layer broadens with downstream distance and consists of a tailward-moving plasma of intermediate density, as required by observations [*Eastman et al.*, 1976; *Paschmann et al.*, 1978; *Eastman and Hones*, 1979; *Sckopke et al.*, 1981; *Mitchell et al.*, 1987] and clearly continues to broaden beyond the exponential growth region, $x = 1$ to 2.5 R_E, identified in Figure 10. The broadening seen for ρ, $\mathbf{v}$ and U_{tot} confirms that the KH instability is indeed transferring mass, momentum and energy across the magnetopause into the magnetosphere. The differences between the three cases show themselves primarily in the rate of development of the boundary layer which slows its broadening as the magnetic field becomes more important in cases B and C, as predicted by the linear stability analysis (see Table 2).

The plot of $\tilde{\mathbf{v}} - \mathbf{v}_0$ in Figure 15 demonstrates the development of a fast boundary layer on the earthward side of the magnetopause due to stresses acting across the magnetopause. As the boundary layer develops, it draws in plasma from the adjacent regions to replace the plasma removed by the downstream-

accelerated flow. Since momentum is conserved and the momentum fluxes through all but the outflow boundary are fixed, these entrainment flows must draw the plasma they need through that boundary, thus leading to the upstream flows seen on the earthward side. For a larger simulation grid, the restrictions imposed on the entrainment flows by the boundaries would be less severe and the entrained flow would be free to come from directions other than downstream. With the current simulation grid, the earthward side of the flow field has the appearance of a sunward-flowing CPS plasma which is slowed as it is entrained into the shear layer and may actually be a realistic flow configuration in spite of the boundary conditions. When active, the entrainment process may reduce the occurrence of magnetic merging on the dayside magnetopause by slowing the CPS plasma enough before it reaches the merging region that it is unable to help drive the merging process.

The boundary layer broadening is due to the action of the time-averaged Reynolds and Maxwell stresses in transporting momentum into the magnetosphere. The Maxwell stress is already present in (5) and becomes identifiable as such after collecting and rewriting the magnetic field terms:

$$\frac{\partial}{\partial t}(\rho v_i)+\frac{\partial}{\partial x_j}(\rho v_i v_j)=-\frac{\partial p}{\partial x_i}+\frac{\partial \tau_{ij}}{\partial x_j}, \tag{20}$$

where

$$\frac{\partial \tau_{ij}}{\partial x_j}=\frac{1}{\mu_0}\left[(\nabla\times\mathbf{B})\times\mathbf{B}\right]_i \tag{21}$$

and τ_{ij} is the magnetic component of the Maxwell stress tensor, $\tau_{ij}=\left[B_iB_j-\frac{1}{2}\mathbf{B}\cdot\mathbf{B}\delta_{ij}\right]/\mu_0$ [e.g., *Jackson*, 1975] and I have adopted the convention of summing over repeated subscripts (e.g., $a_ib_i=a_xb_x+a_yb_y+a_zb_z=\mathbf{a}\cdot\mathbf{b}$). Averaging over (20) gives expressions for the time-averaged Reynolds and Maxwell stresses but first requires a few definitions. The time average of a variable, e.g.,

v_i, is defined by

$$\overline{v}_i = \frac{1}{T_{avg}} \int_{t-\frac{1}{2}T_{avg}}^{t+\frac{1}{2}T_{avg}} v_i \, dt \tag{22}$$

where T_{avg} is large compared to the turbulent time-scale. The definition of the time average may be used to decompose v_i into slowly and quickly varying parts, $v_i = \overline{v}_i + v_i'$. The Favre (density-weighted) average [*Favre*, 1965],

$$\tilde{v}_i = \overline{\rho v_i} / \overline{\rho}, \tag{23}$$

gives another way of decomposing v_i, $v_i = \tilde{v}_i + v_i''$, and permits the effects of compressibility to be included in the Reynolds stress by making an important correlation zero [*Hinze*, 1975]:

$$\begin{aligned} \overline{\rho v_i} &= \overline{\rho(\tilde{v}_i + v_i'')} \\ &= \overline{\rho}\tilde{v}_i + \overline{\rho v_i''} \\ &= \overline{(\overline{\rho} + \rho')}\tilde{v}_i + \overline{\rho v_i''} \\ &= \overline{\rho}\tilde{v}_i + \overline{\rho'}\tilde{v}_i + \overline{\rho v_i''}, \end{aligned} \tag{24}$$

and by using (23) and noting that $\overline{\rho'} = 0$, the correlation $\overline{\rho v_i''}$ must be zero. Time-averaging (20) and substituting using $\rho_i = \overline{\rho}_i + \rho_i'$ and $v_i = \tilde{v}_i + v_i''$ gives

$$\frac{\partial}{\partial t}\left(\overline{(\overline{\rho}_i + \rho')(\tilde{v}_i + v_i'')}\right) + \frac{\partial}{\partial x_j}\left(\overline{(\overline{\rho}_i + \rho')(\tilde{v}_i + v_i'')(\tilde{v}_j + v_j'')}\right) = -\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial \overline{\tau}_{ij}}{\partial x_j}. \tag{25}$$

Expanding the left-hand side, using $\overline{\rho'} = 0$ and $\overline{\rho v_i''} = 0$ (note that $\overline{v_i''} \neq 0$) and rearranging the terms gives the time-averaged equation of motion,

$$\frac{\partial}{\partial t}(\overline{\rho}\tilde{v}_i) + \frac{\partial}{\partial x_j}(\overline{\rho}\tilde{v}_i\tilde{v}_j) = -\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\overline{\tau}_{ij} - \overline{\rho v_i'' v_j''}\right). \tag{26}$$

The new term, $-\overline{\rho v_i'' v_j''}$, is the Reynolds stress. Making the substitution

$$\eta_{ano}\frac{\partial\tilde{v}_i}{\partial x_j}=\overline{\tau}_{ij}-\overline{\rho v_i'' v_j''} \tag{27}$$

allows introduction of an anomalous viscosity, η_{ano} [e.g., *Schlichting*, 1968; *Miura*, 1984], into (20) and gives

$$\frac{\partial}{\partial t}(\overline{\rho}\tilde{v}_i)+\frac{\partial}{\partial x_j}(\overline{\rho}\tilde{v}_i\tilde{v}_j)=-\frac{\partial\overline{p}}{\partial x_i}+\eta_{ano}\frac{\partial^2\tilde{v}_i}{\partial x_j^2} \tag{28}$$

for constant η_{ano}. The x-component of (28) describes the development of the downstream component of the simulation flow

$$\frac{\partial}{\partial t}(\overline{\rho}\tilde{v}_x)+\frac{\partial}{\partial y}(\overline{\rho}\tilde{v}_x\tilde{v}_y)=-\frac{\partial\overline{p}}{\partial x}+\eta_{ano}\frac{\partial^2\tilde{v}_x}{\partial y^2} \tag{29}$$

with the anomalous viscosity given by

$$\eta_{ano}=\left(\overline{B_xB_y}/\mu_0-\overline{\rho v_x'' v_y''}\right)\Big/\frac{\partial\tilde{v}_x}{\partial y}. \tag{30}$$

A positive value of the anomalous viscosity therefore acts diffusively to broaden the average boundary layer and a value for it may be calculated from simulation results.

The Reynolds and Maxwell stresses, their sum and the transverse energy flux (see Appendix A) are shown in Figure 17 for the three cases during the averaging intervals used to produce Figure 15. Surprisingly, the total stress for the fastest growing case, case A, is the smallest of the three cases due to a large reversal near the outflow boundary, indicating that momentum is being transferred back into the magnetosheath flow. This reversal also manifests itself in the corresponding plot of U_{tot} in Figure 15 as a reduction in the rate at which the energy boundary layer broadens; the same plots for cases B and C show steady broadening for the entire length of the simulation.

Cases B and C both show total stresses that are favorable to continuous

momentum transport into the magnetosphere, but due to different mechanisms. The Reynolds stress is responsible for most of the transport in case B, likely through vortex-tilting [e.g., *Horton et al.*, 1987], while the Maxwell stress is responsible in case C, supporting *Miura*'s [1984] findings for temporally developing simulations of the KH instability. Superposition of the shear layer and the magnetopause in case C is responsible for increasing the Maxwell stress by making it possible for the KH instability developing at the magnetopause to compress the tilted field found in the stagnation regions, thus making $B_x B_y$ large

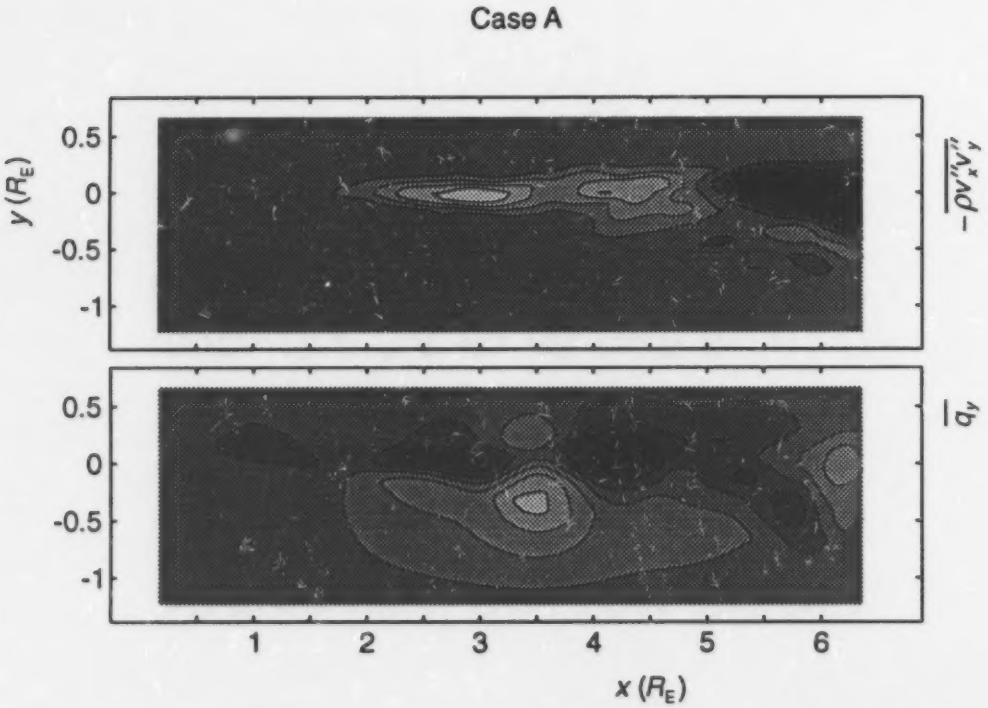
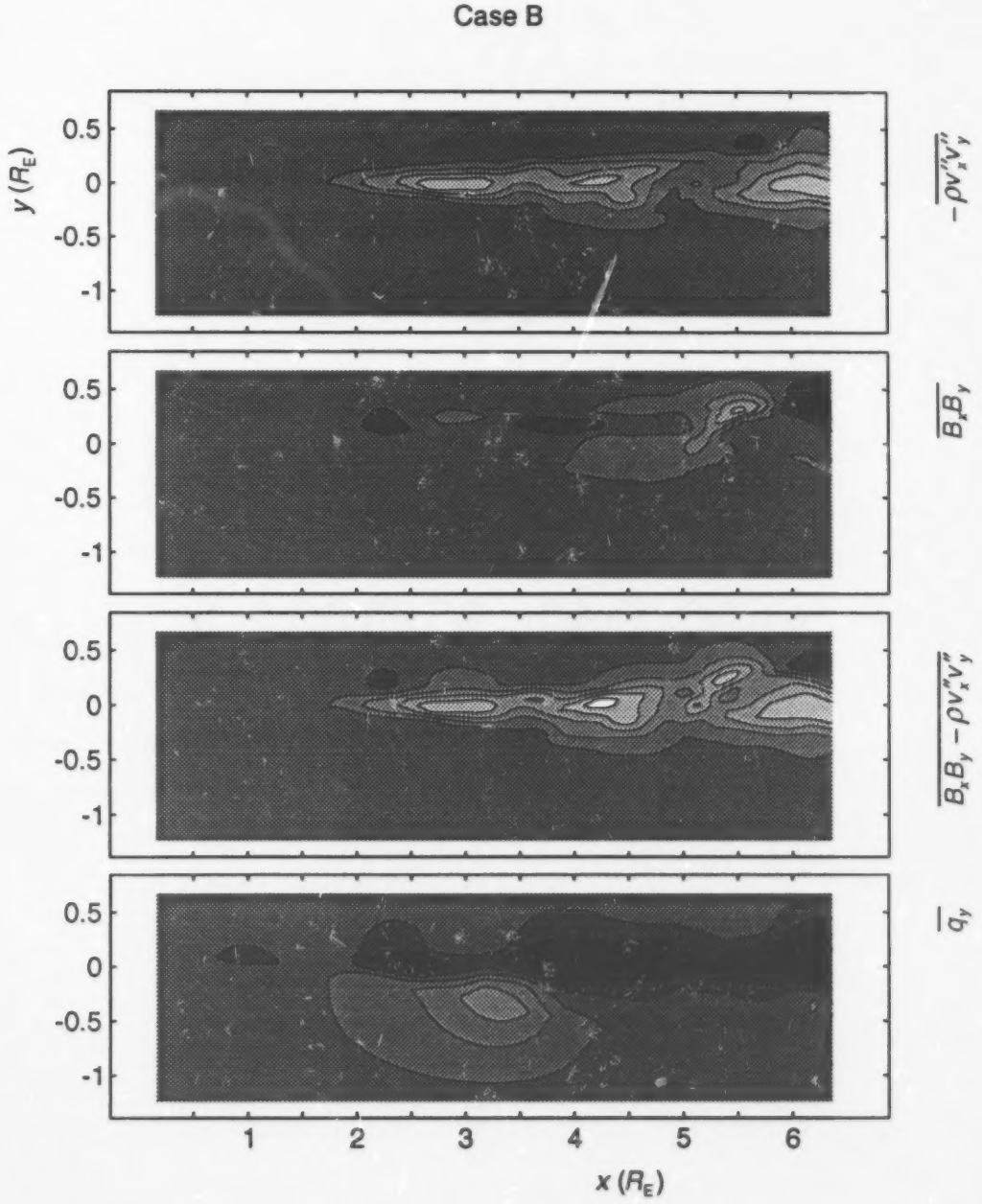


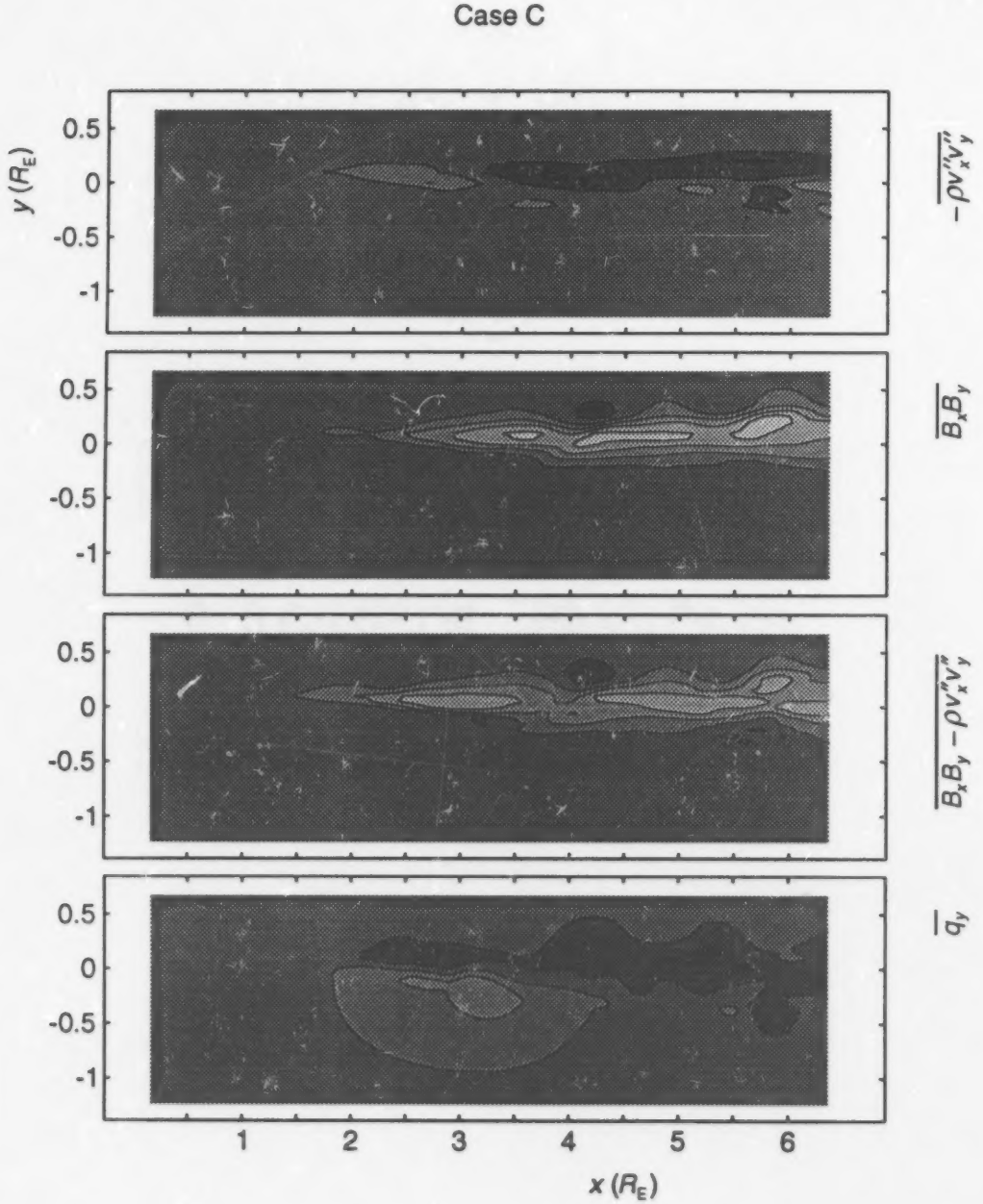
Figure 17. Time-averaged shear stresses and energy flux for the three simulated LLBL configurations. The averaging intervals are the same as used in Figure 15. The contours of stress are at intervals of 1.2×10^{-12} kg · m/s^2, centered about the zero level and the contours of energy flux density are at intervals of 1.7×10^{-6} J/m^2 · s, also centered about the zero level. There is no plot of Maxwell stress for case A because the magnetic field is everywhere directed normal to the plane of the simulation space.

(see case C in Figure 11) and increasing the contribution to the total stress made by the Maxwell stress.

The anomalous viscosity calculated from (30) using the averaged quantities is always larger than the minimum value required to produce the LLBL.



Sonnerup [1980] found that an anomalous kinematic viscosity, $\nu_{ano} = \eta_{ano}/\overline{\rho}$, of 10^9 m^2/s is sufficient to produce the observed boundary layer broadening. The anomalous viscosity found by calculating ν_{ano} from the averages presented in Figures 15 and 17 is everywhere larger than 10^9 m^2/s in the LLBL, from ~2 R_E



downstream of the inflow boundary to the outflow boundary, and is almost everywhere greater than 10^{10} m^2/s for all three cases.

CONCLUSIONS

All of the boundary layer configurations considered in the previous section show the spatial development of a tailward-broadening boundary layer just inside the average position of the magnetopause, the low-latitude boundary layer (LLBL), which is composed of a plasma having density intermediate between the magnetosheath and magnetosphere plasma densities and moving tailward at high speeds in accordance with observations. The broadening may be described as the result of an anomalous viscosity representing the effects of the Reynolds and Maxwell stresses. The values calculated for the anomalous viscosity exceed the value required to produce the observed boundary layer [*Sonnerup*, 1980] everywhere in the LLBL and for all of the configurations considered.

As for conventional fluid viscosity, anomalous viscosity in the LLBL leads to entrainment of the surrounding plasma into the developing boundary layer. Practically all of the plasma entrained comes from the magnetosphere where it convects sunward under the influence of the cross-tail electric field [*Baumjohann et al.*, 1989]. Once in the boundary layer, it is accelerated tailward and returns to the magnetotail from whence it came, perhaps completing a large-scale circulation pattern.

For one of the boundary layer configurations considered, the total stress showed a significant reversal over a large region of the LLBL. In this region, momentum and energy were transferred into the magnetosheath from the magnetosphere in a manner reminiscent of that observed by *Lu and Wu* [1991] in hydrodynamical simulations of a supersonic shear layer. The other boundary layer configurations showed total stresses that were only weakly reversed in very small regions suggesting that the flow-aligned component of the magnetic field present for those configurations was somehow responsible for assuring almost continuous momentum and energy transport into the magnetosphere.

The presence of a flow-aligned component of the magnetic field in the magnetosheath slows the spatial development of the LLBL and reduces mixing between the magnetosheath and magnetosphere plasmas due to the stabilizing effect of magnetic field line tension. A flow-aligned component of the field can also prevent large, fully developed vortices from rolling up beyond their ability to bend the field lines threading them. A vortex in such a situation first ceases its roll-up and then appears to be sheared away from the magnetosphere by the fast magnetosheath flow. The nonlinear stabilization of the boundary layer by the magnetic field makes it unable to continue mixing plasma across the boundary so that less efficient methods of transporting plasma into the magnetosphere once again become important.

The process of wave-particle diffusion frequently invoked as a means of transporting magnetosheath plasma into the magnetosphere [e.g., *LaBelle and Treumann*, 1988] may be aided by the Kelvin-Helmholtz (KH) instability. The large vortices created by the instability act as obstacles to the magnetosheath flow and have magnetosheath plasma pressed against their upstream sides. The sharp gradients that develop between the magnetosheath and magnetosphere plasmas in these regions are steep enough to restart the wave-particle diffusion process assumed to have produced the initial boundary layer. In this way, magnetosheath plasma that would otherwise have been unable to enter the LLBL because it was frozen onto field lines whose magnetic tension prevent them from being entrained can pass into the boundary layer. Although the simulation is at present incapable of treating this diffusion properly, it has permitted identification of regions where it could be important.

Even though the simulation and initial boundary layer configurations have been made as realistic as possible, there are inevitable shortcomings which both aid and hinder the development of the spatial instability. Probably the most significant shortcoming is the simulation's current inability to deal with the curved geometry necessary to include variations in the magnetosheath plasma

parameters along the length of the simulation space. In reality, the magnetosheath velocity increases from zero at the subsolar point to faster than the solar wind speed at the flanks of the magnetosphere with similar large changes occurring to the other plasma parameters [*Spreiter et al.*, 1966]. The changing conditions along the magnetopause have the potential to introduce new possibilities for the development of the LLBL. For example, downstream variations of the boundary layer parameters mean that the wavenumber and frequency of the fastest growing KH instability mode will also vary, thus leading to the possibility that the fastest growing mode at an upstream position will develop and dominate the faster growing downstream modes simply because it has more space in which to develop.

A magnetosheath velocity which increases along the length of the magnetopause would provide a streamwise-increasing source of free energy with which to drive the instability. The current simulation shows that the KH instability in all three cases develops spatial at an exponential rate from the noise level to saturation in approximately 1.5 R_E and then enters a phase of slow growth characterized by vortex merging, diffusive smoothing and steady, linear broadening of the boundary layer. With the continual addition of free energy made possible by an increasing magnetosheath velocity, it is unlikely that the instability would ever truly saturate, but would experience a phase of exponential growth as before and then enter a phase of intermediate growth having some of the characteristics of both the exponential and slow growth phases. However, if, as supposed, the boundary layer is nonlinearly stabilized by a flow-aligned component of the magnetic field, the phase of intermediate growth will be greatly reduced but may also be aided by the downstream weakening of the magnetosheath magnetic field.

Another shortcoming of the simulation is its current inability to include three dimensional effects. The equatorial plane of the LLBL where the initial boundary layer configurations are situated is connected through the field lines

threading it to distant regions of the magnetosphere having rather different plasma and flow parameters. The fastest growing mode at the equatorial plane may not be the fastest growing mode in other regions on the same field lines so the mode in the equatorial plane may develop more slowly than predicted as it loses energy in its attempts to get the other regions developing. On the other hand, the field geometry may be such that a fully-developed instability in one region couples favorably to another region where the instability is barely underway thus greatly accelerating its development.

As well as addressing these shortcomings, future work will explore some of the avenues of inquiry opened by this thesis. For example, a simulated satellite pass through a developing LLBL would yield a "dataset" which could be compared against actual satellite passes to further assess the role of the KH instability in the observed boundary layer. Spatial and temporal energy spectra, as well as other diagnostics available from fluid turbulence theory, would enable examination of the turbulent structure of the boundary layer [*Lesieur et al.*, 1988] and would indicate the degree to which turbulence theory may be used to understand the development and dynamics of the boundary layer. A detailed energy budget of a vortex roll-up in the case of a coincident magnetopause and shear layer (i.e., case C) would help to further understanding of the nonlinear stabilization process. Adding the effects of resistivity to the equations would permit consideration of boundary layer configurations for which the magnetic field rotation at the magnetopause is large enough to introduce the possibility of magnetic merging.

This thesis has presented the results of simulations of the spatial development of the LLBL (summarized in Figure 18) which confirm, once again, the significance of the KH instability in determining the structure and dynamics of the boundary layer. This work has striven to increase the realism of simulations of the LLBL by investigating several realistic boundary layer configurations using an ideal magnetohydrodynamic (MHD) simulation implemented on a

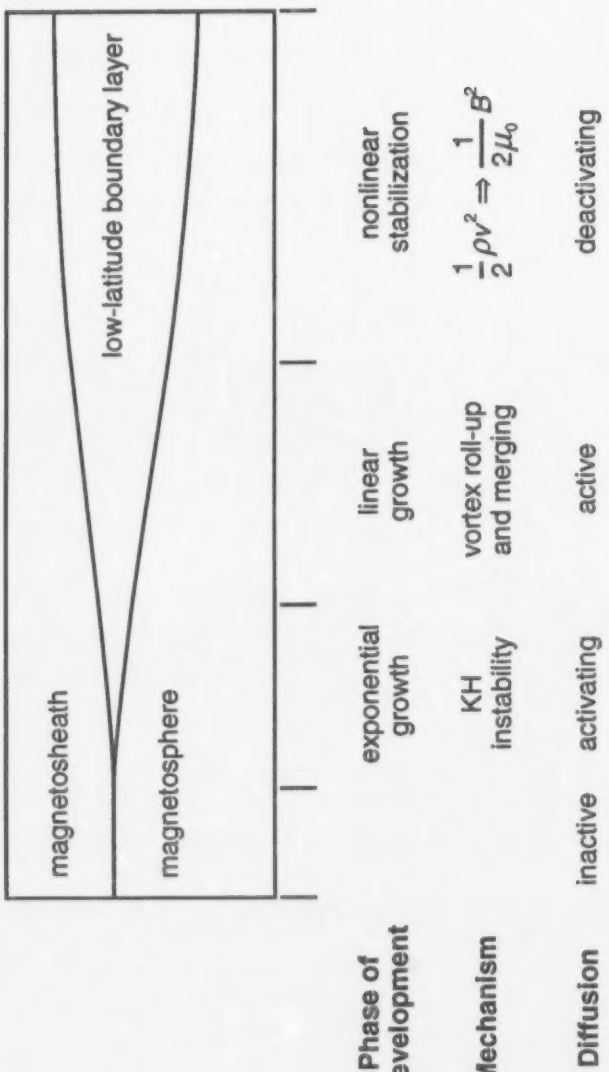


Figure 18. A model of the spatial development of the LLBL based on the three simulated LLBL configurations.

nonperiodic grid with carefully constructed boundary conditions. The boundary layer configurations were chosen to represent possible configurations of an important region of the LLBL while remaining computationally feasible to simulate and extend to three dimensions in later work. The simulation makes no simplifying assumptions about the equations being solved and has been shown to be capable of realistically simulating the ideal MHD equations in the LLBL. The boundary conditions implemented permit the KH instability in the LLBL to be followed well into its quasi-stationary state without compromising the quality of the simulation results.

BIBLIOGRAPHY

Axford, W. I., and C. O. Hines, A unifying theory of high-latitude geophysical phenomena and geomagnetic storms, *Can. J. Phys., 39*, 1433, 1961.

Axford, W. I., Note on a problem of magnetohydrodynamic stability, *Can. J. Phys., 40*, 654, 1962.

Batchelor, G. K., *An Introduction to Fluid Dynamics*, pp. 511, Cambridge University Press, Cambridge, 1967.

Baumjohann, W., G. Paschmann, and C. A. Cattell, Average plasma properties in the central plasma sheet, *J. Geophys. Res., 94*, 6597, 1989.

Belmont, G., and G. Chanteur, Advances in magnetopause Kelvin-Helmholtz instability studies, *Phys. Scri., 40*, 124, 1989.

Berchem, J., and C. T. Russell, The thickness of the magnetopause current layer: ISEE 1 and 2 observations, *J. Geophys. Res., 87*, 2108, 1982.

Berchem, J., and H. Okuda, A two-dimensional particle simulation of the magnetopause current layer, *J. Geophys. Res., 95*, 8133, 1990.

Book, D. L., J. P. Boris, and S. T. Zalesak, Flux-corrected transport, in *Finite-Difference Techniques for Vectorized Fluid Dynamics Calculations*, edited by D. L. Book, pp. 29-55, Springer-Verlag, New York, 1981.

Book, D. L., and M. A. Fry, Airblast simulations using flux-corrected transport codes, in *Computational Techniques and Applications: CTAC-83*, edited by J. Noye and C. Fletcher, pp. 826-840, Elsevier, New York, 1984.

Boris, J. P., and D. L. Book, Flux-corrected transport. III. Minimal-error FCT algorithms, *J. Comp. Phys., 20*, 397, 1976a.

Boris, J. P., Flux-corrected transport modules for solving generalized continuity equations, *NRL Memorandum Report No. 3237*, 79 pp., Naval Research Laboratory, Washington, D. C., 1976b.

Boris, J. P., On large eddy simulations using subgrid turbulence models, in *Whither Turbulence? Turbulence at the Crossroads*, edited by J. L. Lumely, pp. 344-353, Springer-Verlag, New York, 1990.

Brackbill, J. U., and D. C. Barnes, The effect of nonzero $\nabla \cdot \mathbf{B}$ on the numerical solution of the magnetohydrodynamic equations, *J. Comput. Phys., 35*, 426, 1980.

Carpenter, R. L., K. K. Droegemeier, P. R. Woodward, and C. E. Hane, Application of the piecewise parabolic method (PPM) to meteorological modelling, *Mon. Wea. Rev., 118*, 586, 1990.

Celnikier, L. M., L. Muschietti, and M. V. Goldman, Aspects of interplanetary plasma turbulence, *Astron. Astrophys., 181*, 138, 1987.

Chen, X. L., and P. J. Morrison, A sufficient condition for the ideal instability of shear flow with parallel magnetic field, *Phys. Fluids B, 3*, 863, 1991.

Corcos, G. M., and F. S. Sherman, The mixing layer: deterministic models of a turbulent flow. Part 1. Introduction and the two-dimensional flow, *J. Fluid Mechanics, 139*, 29, 1984.

DeVore, C. R., Flux-corrected transport techniques for multidimensional compressible magnetohydrodynamics, *J. Comput. Phys., 92*, 142, 1991.

Drazin, P. G., and W. H. Reid, *Hydrodynamic Stability*, pp. 201, Cambridge University Press, Cambridge, 1981.

Dungey, J. W., Interplanetary magnetic field and the auroral zones, *Phys. Rev. Lett., 6*, 47, 1961.

Eastman, T. E., E. W. Hones, Jr., S. J. Bame, and J. R. Ashbridge, The magnetospheric boundary layer: site of plasma, momentum and energy transfer from the magnetosheath into the magnetosphere, *Geophys. Res. Letters, 3*, 685, 1976.

Eastman, T. E., and E. W. Hones, Jr., Characteristics of the magnetospheric boundary layer and magnetopause layer as observed by Imp 6, *J. Geophys. Res., 84*, 2019, 1979.

Eastman, T. E., B. Popielawska, and L. A. Frank, Three-dimensional plasma observations near the outer magnetospheric boundary, *J. Geophys. Res., 90*, 9519, 1985.

Engebretson, M. J., N. Lin, W. Baumjohann, H. Luehr, B. J. Anderson, L. J. Zanetti, T. A. Potemra, R. L. McPherron, and M. G. Kivelson, A comparison of ULF fluctuations in the solar wind, magnetosheath, and dayside magnetosphere 1. magnetosheath morphology, *J. Geophys. Res., 96*, 3441, 1991.

Favre, A., Equations des gaz turbulents compressibles, *J. de Mécanique, 4*, 361, 1965.

Fletcher, C. A. J., *Computational Techniques for Fluid Dynamics, Vol. 1*, Springer-Verlag, New York, 1988.

Fujimoto, M., and T. Terasawa, Ion inertia effect on the Kelvin-Helmholtz instability, *J. Geophys. Res., 96*, 15,725, 1991.

Gary, S. P., and A. G. Sgro, The lower hybrid drift instability at the magnetopause, *Geophys. Res. Lett., 17*, 909, 1990.

Givoli, D., Non-reflecting boundary conditions, *J. Comput. Phys., 94*, 1, 1991.

Grandjouan, N., The modified equation approach to flux-corrected transport, *J. Comput. Phys., 91*, 424, 1990.

Grinstein, F. F., E. S. Oran, and J. P. Boris, Pressure field, feedback, and global instabilities of subsonic spatially developing mixing layers, *Phys. Fluids A, 3*, 2401, 1991.

Grinstein, F. F., and R. H. Guirguis, Effective viscosity in the simulation of spatially evolving shear flows with monotonic FCT models, *J. Comput. Phys., 101*, 165, 1992.

Hinze, J. O., *Turbulence*, pp. 20, McGraw-Hill, New York, 1975.

Hirsch, C., *Numerical Computation of Internal and External Flows, Vol. 2*, pp. 528, Wiley, New York, 1988.

Hones, E. W., Jr., J. R. Asbridge, S. J. Bame, M. D. Montgomery, S. Singer, and S.-I. Akasofu, Measurement of magnetotail plasma flow made with Vela 4B, *J. Geophys. Res., 77*, 5503, 1972.

Horton, W., T. Tajima, and T. Kamimura, Kelvin-Helmholtz instability and vortices in magnetized plasma, *Phys. Fluids, 30*, 3485, 1987.

Ichimaru, S., *Basic Principles of Plasma Physics*, pp. 290, W. A. Benjamin, London, 1973.

Jackson, J. D., *Classical Electrodynamics*, 2nd ed., pp. 239, Wiley, Toronto, 1975.

Kaul, U. K., Do large structures control their own growth in a mixing layer? An assessment, *J. Fluid Mechanics, 190*, 427, 1988.

LaBelle, J., and R. A. Treumann, Plasma waves at the dayside magnetopause, *Space Science Reviews, 47*, 175, 1988.

Lemaire, J., and M. Roth, Penetration of solar wind plasma elements into the magnetosphere, *J. Atmos. Terr. Phys., 40*, 331, 1978.

Lesieur, M., C. Staquet, P. Le Roy, and P. Comte, The mixing layer and its coherence examined from the point of view of two-dimensional turbulence, *J. Fluid Mechanics, 192*, 511, 1988.

Lu, P. J., and K. C. Wu, Numerical investigation on the structure of a confined supersonic mixing layer, *Phys. Fluids A, 3*, 3063, 1991.

Lundin, R., On the magnetospheric boundary layer and solar wind energy transfer into the magnetosphere, *Space Sci. Rev., 48*, 263, 1988.

Lysak, R. L., Electrodynamic coupling of the magnetosphere and ionosphere, *Space Sci. Rev., 52*, 33, 1990.

Matthaeus, W. H., L. W. Klien, S. Gosh, and M. R. Brown, Nearly incompressible magnetohydrodynamics, pseudosound, and solar wind fluctuations, *J. Geophys. Res., 96*, 5421, 1991.

McComb, W. D., *The Physics of Fluid Turbulence*, pp. 436, Oxford University Press, Oxford, 1990.

Mitchell, D. G., F. Kutchko, D. J. Williams, T. E. Eastman, L. A. Frank and C. T. Russell, An extended study of the low-latitude boundary layer on the dawn and dusk flanks of the magnetopause, *J. Geophys. Res., 92*, 7394, 1987.

Mitchell, D. G., M. J. Engebretson, D. J. Williams, C. A. Cattell, and R. Lundin, Pc 5 pulsations in the outer dawn magnetosphere seen by ISEE 1 and 2, *J. Geophys. Res., 95*, 967, 1990.

Miura, A., and P. L. Pritchett, Nonlocal stability analysis of the MHD Kelvin-Helmholtz instability in a compressible plasma, *J. Geophys. Res., 87*, 7431, 1982.

Miura, A., Anomalous transport by magnetohydrodynamic Kelvin-Helmholtz instabilities in the solar wind-magnetosphere interaction, *J. Geophys. Res., 89*, 801, 1984.

Miura, A., Simulation of Kelvin-Helmholtz instability at the magnetospheric boundary, *J. Geophys. Res., 92*, 3195, 1987.

Miura, A., Kelvin-Helmholtz instability for supersonic shear flow at the magnetospheric boundary, *Geophys. Res Lett., 17*, 749, 1990.

Miura, A., Kelvin-Helmholtz instability at the magnetospheric boundary: dependence on the magnetosheath sonic mach number, *J. Geophys. Res., 97*, 10,655, 1992.

Moffatt, H. K., *Magnetic Field Generation in Electrically Conducting Fluids*, pp. 53-62, Cambridge University Press, Cambridge, 1978.

Odstrcil, D., A new optimized FCT algorithm for shock wave problems, *J. Comput. Phys., 91*, 71, 1990.

Ogilvie, W. O., and R. J. Fitzenreiter, The Kelvin-Helmholtz instability at the magnetopause and inner boundary layer surfaces, *J. Geophys. Res., 94*, 15,113, 1989.

Ong, R. S. B., and N. Roderick, On the Kelvin-Helmholtz instability of the Earth's magnetopause, *Planet. Space Sci., 20*, 1, 1972.

Oran, E. S., and J. P. Boris, *Numerical Simulation of Reactive Flow,* pp. 294, Elsevier, New York, 1987.

Ottino, J. M., Kinematics of chaotic mixing: experimental and computational results, in *Topological Fluid Mechanics: Proceedings of the IUTAM Symposium*, edited by H. K. Moffatt and A. Tsinober, pp. 13-22, Cambridge University Press, Cambridge, 1990.

Panides, E., and R. Chevray, Vortex dynamics in a plane, moderate-Reynolds-number shear layer, *J. Fluid Mech., 214*, 411, 1990.

Parks, G. K., *Physics of Space Plasmas*, pp. 162, Addison-Wesley, Redwood City, California, 1991.

Paschmann, G., N. Sckopke, G. Haerendel, J. Papamastorakis, S. J. Bame, J. R. Asbridge, J. T. Gosling, E. W. Hones, and E. R. Tech, ISEE plasma observations near the subsolar magnetopause, *Space Sci. Rev., 22*, 717, 1978.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, pp. 582, Cambridge University Press, Cambridge, 1986.

Pritchett, P. L., and F. V. Coroniti, The collisionless macroscopic Kelvin-Helmholtz instability, 1, Transverse electrostatic mode, *J. Geophys. Res., 89*, 168, 1984.

Rajaram, R., D. G. Sibeck, and R. W. McEntire, Linear theory of the Kelvin-Helmholtz instability in the low-latitude boundary layer, *J. Geophys. Res., 96*, 9615, 1991.

Rayleigh, Lord, On the stability, or instability, of certain fluid motions, *Proc. London Math. Soc., 11*, 57, 1880.

Reynolds, W. C., The potential and limitations of direct and large eddy simulations, in *Whither Turbulence? Turbulence at the Crossroads*, edited by J. L. Lumely, pp. 313-342, Springer-Verlag, New York, 1990.

Rosenhead, L., The formation of vortices from a surface of discontinuity, *Proc. R. Soc. A, 134*, 170, 1931.

Roth, M., On impulsive penetration of solar wind plasmoids into the geomagnetic field, *Planet. Space Sci., 40*, 193, 1992.

Russell, C. T., and R. C. Elphic, ISEE observations of flux transfer events at the dayside magnetopause, *Geophys. Res. Letters, 6*, 33, 1979.

Samson, J. C., B. G. Harrold, J. M. Ruohoniemi, R. A. Greenwald, and A. D. M. Walker, Field line resonances associated with MHD waveguides in the magnetosphere, *Geophys. Res. Letters, 19*, 441, 1992.

Sanderson, T. R., P. Daly, K.-P. Wenzel, E. W. Hones Jr., and E. J. Smith, Observations of a large-scale vortex-like structure in the deep-tail plasma sheet boundary layer, in *Solar Wind-Magnetosphere Coupling*, edited by Y. Kamide and J. A. Slavin, pp. 739-745, Terra Scientific, Tokyo, 1986.

Sandham, N. D., and W. C. Reynolds, Some inlet-plane effects on the numerically simulated spatially-developing mixing layer, *Turbulent Shear Flows 6*, edited by J.-C. André, J. Cousteix, F. Durst, B. E. Launder, F. W. Schmidt, and J. H. Whitelaw, pp. 441-454, Springer-Verlag, Berlin, 1989a.

Sandham, N. D., and W. C. Reynolds, Compressible mixing layer: linear theory and direct simulation, *AIAA Journal, 28*, 618, 1989b.

Schlichting, H., *Boundary Layer Theory*, 6th ed., pp. 544, McGraw-Hill, New York, 1968.

Schmidt, G., *Physics of High Temperature Plasmas*, 2nd ed., pp. 89, Academic Press, New York, 1979.

Sckopke, N., G. Paschmann, G. Haerendel, B. U. Ö., Sonnerup, S. J. Bame, T. G. Forbes, E. W. Hones, Jr. and C. T. Russell, Structure of the low-latitude boundary layer, *J. Geophys. Res., 86*, 2099, 1981.

Shivamoggi, B. K., Magnetohydrodynamic theories of magnetic field reconnection, *Phys. Rep., 127*, 99, 1985.

Sibeck, D. G., R. E. Lopez, and E. C. Roelof, Solar wind control of the magnetopause shape, location, and motion, *J. Geophys. Res., 96*, 5489, 1991.

Sibeck, D. G., Transient events in the outer magnetosphere: boundary waves or flux transfer events?, *J. Geophys. Res., 97*, 4009, 1992.

Sod, G. A., A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, *J. Comput. Phys., 27*, 1, 1978.

Song, P., and C. T. Russell, Model of the formation of the low-latitude boundary layer for strongly northward interplanetary magnetic field, *J. Geophys. Res., 97*, 1411, 1992.

Sonnerup, B. U. Ö., Theory of the low-latitude boundary layer, *J. Geophys. Res., 85*, 2017, 1980.

Sonnerup, B. U. Ö., I. Papamastorakis, G. Paschmann, and H. Lühr, The magnetopause for large magnetic shear: analysis of convection electric fields from AMPTE/IRM, *J. Geophys. Res., 95*, 10,541, 1990.

Southwood, D. J., Magnetopause Kelvin-Helmholtz instability, in *Magnetospheric Boundary Layers*, edited by B. Battrick and J. Mort, pp. 357-364, ESA Sci. Tech. Publ. Branch, Noordwijk, The Netherlands, 1979.

Southwood, D. J., and M. G. Kivelson, Magnetospheric interchange instability, *J. Geophys. Res., 92*, 1987.

Spreiter, J. R., A. L. Summers, and A. Y. Alksne, Hydromagnetic flow around the magnetosphere, *Planet. Space Sci., 14*, 1966.

Thompson, K. W., Time-dependent boundary conditions for hyperbolic systems, II, *J. Comput. Phys., 89*, 439, 1990.

Tsurutani, B. T., and R. M. Thorne, Diffusion processes in the magnetopause boundary layer, *Geophys. Res Lett., 9*, 1247, 1982.

Tsurutani, B. T., and T. T. von Rosenvinge, ISEE-3 distant geotail results, *Geophys. Res Lett., 11*, 1027, 1984.

Verzariu, P., Reflection and refraction of hydromagnetic waves at the magnetopause, *Planet. Space Sci., 21*, 2213, 1973.

Walker, A. D. M., The Kelvin-Helmholtz instability in the low-latitude boundary layer, *Planet. Space Sci., 29*, 1119, 1981.

Wei, C. Q., L. C. Lee, and A. L. La Belle-Hamer, A simulation study of the vortex structure in the low-latitude boundary layer, *J. Geophys. Res., 95*, 20,793, 1990.

Woodward, P., and P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, *J. Comp. Phys., 54*, 115, 1984.

Wu, C. C., Kelvin-Helmholtz instability at the magnetopause boundary, *J. Geophys. Res., 91*, 3042, 1986.

Zalesak, S. T., Fully multidimensional flux-corrected transport algorithms for fluids, *J. Comput. Phys., 31*, 335, 1979.

Zalesak, S. T., High order "ZIP" differencing of convective terms, *J. Comput. Phys., 40*, 497, 1981.

Zhuang, M., P. E. Dimotakis, and T. Kubota, The effect of walls on a spatially growing supersonic shear layer, *Phys. Fluids A, 2*, 599, 1990.

APPENDIX A

SOLAR WIND ENERGY INCIDENT ON THE MAGNETOSPHERE

The total energy flux density in an magnetohydrodynamic MHD plasma is given by

$$\mathbf{q} = \left(\frac{1}{2}\rho v^2 + \frac{\gamma}{\gamma - 1} p\right)\mathbf{v} + \frac{1}{\mu_0}\mathbf{E}\times\mathbf{B}, \tag{A.1}$$

where the first term in the parenthesis is the kinetic energy and the second is the combined internal energy and pressure work and the term outside is the Poynting flux. Assuming infinite conductivity, $\mathbf{E} = -\mathbf{v}\times\mathbf{B}$, and $\mathbf{B}$ perpendicular to $\mathbf{v}$ gives

$$\mathbf{q} = \left(\frac{1}{2}\rho v^2 + \frac{\gamma}{\gamma - 1} p + \frac{1}{\mu_0} B^2\right)\mathbf{v}. \tag{A.2}$$

With a few simplifying assumptions about the solar wind and magnetosphere, I can obtain an estimate of the total amount of energy incident on the magnetosphere. Replacing the magnetosphere with a circular disk which is oriented normal to the solar wind and has a radius, R_{msp}, equal to the radius of the magnetosphere gives a total energy flux

$$P = q\pi R_{msp}^2. \tag{A.3}$$

With values typical of the solar wind plasma, i.e., $n = 10\ \text{cm}^{-3}$, $p = 0.003$ nPa, $B = 6.0$ nT and $v = 350$ km/s, the total solar wind energy flux incident on a magnetosphere 20 R_E in radius is 19×10^{12} J/s. For comparison, the solar radiation energy flux incident on a disk one R_E in radius is $170{,}000\times10^{12}$ J/s, some 9000 times greater.

APPENDIX B

REDUCING DISPERSION IN A TIME-CENTERED FLUX-CORRECTED TRANSPORT ALGORITHM[3]

INTRODUCTION

The flux-corrected transport (FCT) technique was developed to deal with problems that arise in the numerical solution of the one-dimensional continuity equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} = 0, \tag{B.1}$$

where ρ is density and u is velocity. The most serious problems arise when the numerical method being used to solve (B.1) is asked to convect a sharp feature in ρ. Most simple algorithms have difficulties in dealing with the short wavelengths associated with such features and produce unwanted oscillations in the vicinity of steep gradients which must somehow be removed (see Figure B.1). Many algorithms add a dissipative term or use artificial viscosity to keep the spurious short-wavelength features from becoming too large. More recent algorithms, such as those based on the FCT [*Boris*, 1976a; *Zalesak*, 1979], piecewise parabolic method [*Carpenter et al.*, 1990] and total variation diminishing [*Hirsch*, 1988] schemes, use nonlinear techniques to inhibit the formation of such features and represent the state of the art in finite difference algorithms.

The most widely used implementation of the FCT technique, the ETBFCT algorithm [*Boris*, 1976b; *Book et al.*, 1981], still suffers from residual errors which result in the formation of "terraces" on steep gradients in ρ. These terraces

---

[3] A version of this appendix has been submitted to the *Journal of Computational Physics*.

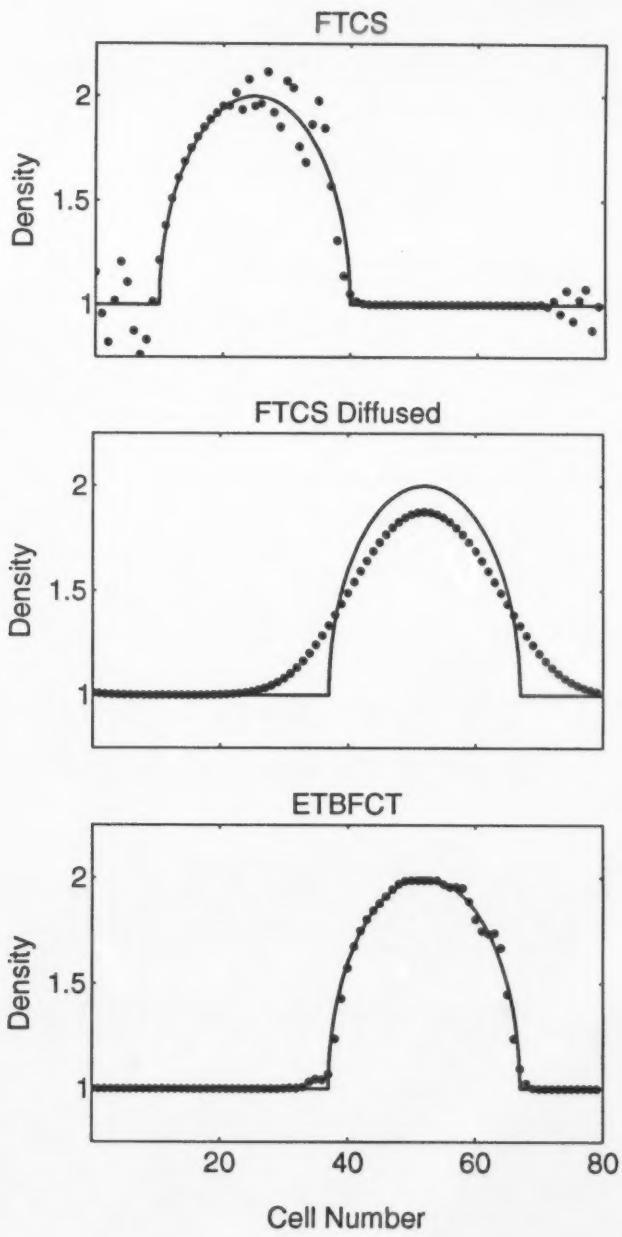


Figure B.1. A semicircular density profile after being convected by three algorithms. The FTCS algorithm is unstable so the profile has been convected only 9 cells whereas the other algorithms are stable and have convected the profile 36 cells across the grid. (Courant number = 0.25)

arise from an interaction between the dispersion error inherent in the algorithm and the nonlinear limiter around which FCT algorithms are built. Although they are usually only cosmetic in nature, occasionally the appearance of terraces will lead to serious errors in a simulation. The frequency of these errors prompted a detailed examination of the original ETBFCT algorithm which has led to the improvements presented here.

In this appendix, I describe the ETBFCT algorithm and apply an analytical technique to study its dispersion and stability characteristics. I then describe *Book and Fry*'s [1984] dispersion-reducing extension to the algorithm and show how a very simple alteration of their algorithm reduces the dispersion error still further while introducing an element of flexibility. The stability characteristics of both algorithms are examined in greater detail than before by considering time-centered formulations of the algorithms. Finally, I describe three tests of this improved algorithm and present the results of those tests.

SINGLE STEP ALGORITHMS

The simplest method of solving (B.1) is by finite differences. For example, a simple finite difference form for (B.1), with u constant, is

$$\frac{\rho_i^{n+1} - \rho_i^n}{\Delta t} + \frac{\rho_{i+1}^n u - \rho_{i-1}^n u}{2\Delta x} = 0, \tag{B.2}$$

where Δt and Δx are the numerical timestep and spatial step sizes, respectively. The superscripts identify different time levels, $t = n\Delta t$, while the subscripts identify different spatial positions, $x = i\Delta x$. This particular difference scheme (Forward-Time Centered-Space or FTCS) is unconditionally unstable for all Δt and requires the addition of a diffusion term for stability to be possible. Even with enough diffusion to guarantee stability, dispersion problems make the FTCS scheme unsuitable for serious use. Stability and dispersion problems such as these are present in all finite difference schemes. The degree to which these

problems can be minimized determines the usefulness of an algorithm.

The ETBFCT algorithm [*Boris*, 1976b; *Book et al.*, 1981] adds corrective fluxes to the FTCS scheme which serve to reduce the algorithm's stability and dispersion problems. Writing (B.2) as

$$\left[\rho_i^{n+1}\right]_{FTCS} = \rho_i^n - \frac{\left[F_{i+\frac{1}{2}}^n\right]_T - \left[F_{i-\frac{1}{2}}^n\right]_T}{\Delta x} \tag{B.3}$$

where

$$\left[F_{i+\frac{1}{2}}^n\right]_T = \frac{1}{2} u \Delta t \left(\rho_{i+1}^n + \rho_i^n\right) \tag{B.4}$$

puts (B.2) into a form which makes the action of the fluxes clearer. $\left[F_{i\pm\frac{1}{2}}^n\right]_T$ are the transport flux terms and are responsible for convective transport of ρ through the walls of the ith numerical cell at $x = (i \pm 1/2)\Delta x$. The finite difference method used in ETBFCT adds two more fluxes to the simple formulation given in (B.3). The first diffuses the FTCS solution enough to guarantee stability and remove all spurious oscillations, but does so at the expense of sharp gradients. The second flux antidiffuses the diffused FTCS solution to bring back the gradients which were smoothed away by the diffusive flux (see Figure B.1). Before the antidiffusive flux is applied to the diffused FTCS solution, it is first limited by a FCT limiter to prevent the numerical oscillations from reforming. The difference equation used in ETBFCT is then

$$\rho_i^{n+1} = \left[\rho_i^{n+1}\right]_{FTCS} + \frac{\left[F_{i+\frac{1}{2}}^n\right]_D - \left[F_{i-\frac{1}{2}}^n\right]_D}{\Delta x} - \frac{\left[F_{i+\frac{1}{2}}^{n+1}\right]_A - \left[F_{i-\frac{1}{2}}^{n+1}\right]_A}{\Delta x}, \tag{B.5}$$

where

$$\left[F_{i+\frac{1}{2}}^n\right]_D = \nu_p \Delta x \left(\rho_{i+1}^n - \rho_i^n\right) \text{ and } \left[F_{i+\frac{1}{2}}^{n+1}\right]_A = \text{Limit}\left[\mu_p \Delta x \left(\left[\rho_{i+1}^{n+1}\right]_{FTCS} - \left[\rho_i^{n+1}\right]_{FTCS}\right)\right] \tag{B.6}$$

are the diffusive and antidiffusive fluxes and ν_p and μ_p are the diffusion and

antidiffusion coefficients, respectively. The limiter used by ETBFCT, represented by Limit[ ] in (B.6), and most other implementations of FCT is described by *Boris* [1976b] and *Book et al.* [1981].

The diffusion and antidiffusion coefficients are chosen to minimize ETBFCT's stability and dispersion problems. One way of quantifying these errors is through application of Von Neumann's method of stability analysis [e.g., *Fletcher*, 1988]. The method examines the amplitude and phase of a single Fourier mode as it is advanced in time by the finite difference scheme under test. Substituting for ρ_i^n in the difference scheme using

$$\rho_i^n = r^n e^{Ikx} = r^n e^{Ik\Delta x} = r^n e^{I\beta}, \tag{B.7}$$

where $I = \sqrt{-1}$, r^n is the amplitude of the mode with wave number k and $\beta = k\Delta x$, gives an expression for ρ_i^{n+1}. The amplitude error of the scheme is then the relative change in the amplitude of ρ_i from one timestep to the next,

$$A = |G| - 1, \tag{B.8}$$

where $G = \rho_i^{n+1}/\rho_i^n$. The phase error of the scheme is the relative phase shift that develops between the solution generated by the difference scheme and the exact solution during a timestep:

$$R = \frac{\xi - u\Delta t}{u\Delta t} = \frac{k\xi}{\varepsilon\beta} - 1, \tag{B.9}$$

where x is the distance that the mode is propagated by the scheme in time Δt and ε is the Courant number, $\varepsilon = u\,\Delta t/\Delta x$. The value of $k\xi$ is related to G through $\tan k\xi = -\mathrm{Im}\,G/\mathrm{Re}\,G$. Applying Von Neumann's method to (B.5) and then expanding the amplitude and phase errors in terms of β gives:

$$A = \left[2(\mu_p - \nu_p) + \varepsilon^2\right]\beta^2 + \left[(\mu_p - \nu_p)\left(\mu_p - \nu_p - \frac{1}{6}\right) + \left(2\mu_p - \frac{1}{3}\right)\varepsilon^2\right]\beta^4 + O(\beta^6), \tag{B.10}$$

the amplitude error and

$$R = \left[-\frac{1}{6} + \nu_p - \frac{1}{3}\varepsilon^2\right]\beta^2 + \left[\frac{1}{120} - \nu_p\left(\mu_p - \nu_p + \frac{1}{4}\right) + \left(\frac{1}{6} - \nu_p\right)\varepsilon^2 + \frac{1}{5}\varepsilon^4\right]\beta^4 + O(\beta^6), \tag{B.11}$$

the phase error. The linear Von Neumann method cannot deal with the nonlinear Limit[ ] operation applied in determining $\left[F_{i+\frac{1}{2}}^{n+1}\right]_A$ and must be replaced by the identity operator while performing the stability analysis.

Appropriate choices for ν_p and μ_p will minimize the algorithm's amplitude and phase error. The values originally used with ETBFCT [*Boris*, 1976b],

$$\nu_p = \frac{1}{6} + \frac{1}{3}\varepsilon^2 \text{ and } \mu_p = \frac{1}{6} - \frac{1}{6}\varepsilon^2, \tag{B.12}$$

were chosen so as to reduce both sources of error to fourth order (see Figure B.2). The error that remains, however, is still sufficient to create the "terraces" that are the bane of FCT algorithms. Terracing is the result of residual phase errors which cause the short wavelength modes present at a sharp edge to lag behind the edge as it propagates across the grid. The limiter sees these modes as newly created peaks which it tries to remove but instead merely flattens into plateaus or, if the new peaks form in the vicinity of a sharp gradient, terraces as in Figure B.1.

It is possible to reduce the phase error still further by adding another source

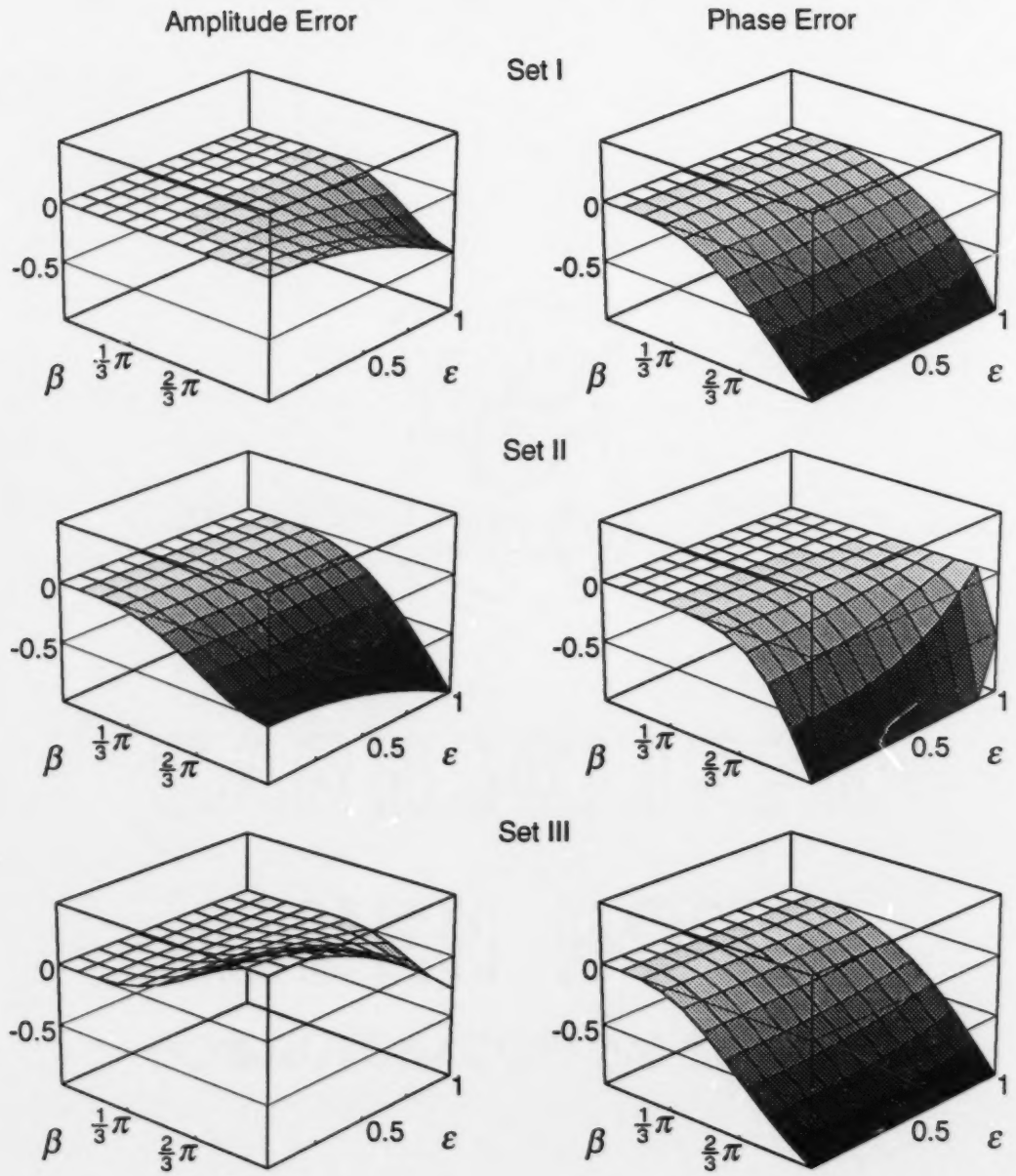


Figure B.2. The exact amplitude and phase error of the single step ETBFCT algorithm for three sets of coefficients: Set I is Boris's original set, Set II is Book and Fry's extended phase accurate set and Set III (with $\nu_p = 0.36467$) is this appendix's tunable set (see Table B.1). The Courant number, e, has been doubled to aid comparison with Figure B.3.

of diffusion [*Book and Fry*, 1984]. Prediffusing the FTCS solution used in ETBFCT introduces a new diffusion coefficient, γ_p, which may be set so as to reduce either the amplitude or dispersion error (but not both) to sixth order:

$$\left[\rho_i^{n+1}\right]_{FTCSD} = \left[\rho_i^{n+1}\right]_{FTCS} + \frac{\left[F_{i+\frac{1}{2}}^n\right]_{preD} - \left[F_{i-\frac{1}{2}}^n\right]_{preD}}{\Delta x}, \tag{B.13}$$

where $\left[F_{i+\frac{1}{2}}^n\right]_{preD} = \gamma_p \Delta x\left(\rho_{i+1}^n - \rho_i^n\right)$. The extended ETBFCT algorithm that results is identical to the original apart from the FTCSD (FTCS Diffused) solution which now replaces the FTCS solution everywhere in (B.5) and (B.6). Performing a Von Neumann analysis on this, Book and Fry's, extended ETBFCT algorithm gives an amplitude error

$$\begin{aligned} A = &\left[2\left(\mu_p - \nu_p - \gamma_p\right) + \varepsilon^2\right]\beta^2 \\ &+\left[-2\gamma_p\mu_p + \left(\mu_p - \nu_p - \gamma_p\right)\left(\mu_p - \nu_p - \gamma_p - \frac{1}{6}\right) + \left(2\mu_p - \frac{1}{3}\right)\varepsilon^2\right]\beta^4 \\ &+O\left(\beta^6\right) \end{aligned} \tag{B.14}$$

and a phase error

$$\begin{aligned} R = &\left[-\frac{1}{6} + \nu_p + \gamma_p - \frac{1}{3}\varepsilon^2\right]\beta^2 \\ &+\left[\frac{1}{120} - \nu_p\mu_p + \left(\nu_p + \gamma_p\right)\left(\nu_p + \gamma_p - \frac{1}{4}\right) + \left(\frac{1}{6} - \nu_p - \gamma_p\right)\varepsilon^2 + \frac{1}{5}\varepsilon^4\right]\beta^4 \\ &+O\left(\beta^6\right). \end{aligned} \tag{B.15}$$

Choosing

$$\nu_p = \frac{1}{6} + \frac{1}{12}\varepsilon^2,\ \mu_p = \frac{1}{6} - \frac{1}{6}\varepsilon^2 \text{ and } \gamma_p = \frac{1}{4}\varepsilon^2 \tag{B.16}$$

reduces the algorithm's amplitude error to sixth order while choosing

$$\nu_p = -\frac{1}{30} + \frac{2}{15}\varepsilon^2,\ \mu_p = \frac{1}{6} - \frac{1}{6}\varepsilon^2 \text{ and } \gamma_p = \frac{1}{5} + \frac{1}{5}\varepsilon^2 \qquad \text{(B.17)}$$

reduces its phase error to sixth order.

Unfortunately, both sets of coefficients have their problems. The amplitude accurate set given in (B.16) results in an algorithm which still suffers from phase error as large as that of the original ETBFCT algorithm. Even the residual amplitude error for this set of coefficients is not significantly improved from that of the original set given in (B.12) (with $\gamma_p = 0$).

The phase accurate set, (B.17), gives the algorithm very good phase properties, but at the expense of amplitude accuracy (see Figure B.2). The short wavelength modes are strongly filtered as a result of the relatively poor amplitude accuracy of the algorithm (although it is still fourth order accurate). The combination of filtering at short wavelengths and good phase properties is desirable for many problems: the filtering removes those modes which could, if permitted to persist, eventually suffer from phase errors. However, a slowly evolving system simulated using such an algorithm may suffer from excessive filtering because the filter will erode stationary and slow-moving structures: a shear flow subjected to the Kelvin-Helmholtz instability is one such system.

TWO STEP ALGORITHMS

All of the algorithms presented in the previous section were assumed to perform their time advancement in a single timestep. In fact, most finite difference methods use some sort of multiple step time advancement scheme to time-center themselves. ETBFCT and other recent efforts made towards improving it [*Odstrcil*, 1990; *Grandjouan*, 1990] are no exception. ETBFCT puts the difference equation described in (B.5) into a predictor-corrector framework in which the solution obtained during the predictor step is used to calculate the transport and antidiffusive fluxes required by the corrector step. A Von Neumann analysis of the full ETBFCT algorithm including both the predictor and corrector

steps shows that the coefficients given in (B.12), coefficients which work well in a single timestep implementation of the algorithm, make the amplitude and phase of the corrector step only second order accurate. Allowing the coefficients to differ between the predictor and corrector steps and choosing

$$\nu_c = \frac{1}{6} - \frac{\varepsilon^2}{24} \text{ and } \mu_c = \frac{1}{6} - \frac{\varepsilon^2}{24}, \tag{B.18}$$

where the subscript c signifies coefficients that are to be used during the corrector step, increases the amplitude accuracy of the corrector step, and therefore the algorithm as a whole, to sixth order and its phase accuracy to fourth order. Unfortunately, a large amount of residual phase error couples with the very good amplitude behavior to make the algorithm very susceptible to terracing errors as well as being weakly unstable to long wavelength modes.

The full predictor-corrector formulation of Book and Fry's extension to ETBFCT results in an algorithm with six coefficients. The availability of such a large number of coefficients makes it possible to build very good amplitude and phase properties into the algorithm. The finite difference equations composing the full ETBFCT algorithm with Book and Fry's extension are, for the predictor step,

$$\rho_i^{n+\frac{1}{2}} = \left[\rho_i^{n+\frac{1}{2}}\right]_{FTCSD} + \frac{\left[F_{i+\frac{1}{2}}^n\right]_D - \left[F_{i-\frac{1}{2}}^n\right]_D}{\Delta x} - \frac{\left[F_{i+\frac{1}{2}}^{n+\frac{1}{2}}\right]_A - \left[F_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right]_A}{\Delta x} \tag{B.19}$$

where

$$\left[\rho_i^{n+\frac{1}{2}}\right]_{FTCSD} = \rho_i^n - \frac{\left[F_{i+\frac{1}{2}}^n\right]_T - \left[F_{i-\frac{1}{2}}^n\right]_T}{\Delta x} + \frac{\left[F_{i+\frac{1}{2}}^n\right]_{preD} - \left[F_{i-\frac{1}{2}}^n\right]_{preD}}{\Delta x} \tag{B.20}$$

with the fluxes

$$
\begin{aligned}
\left[F^n_{i+1/2}\right]_T &= \frac{1}{2}u^n_{i+1/2}(\Delta t/2)\left(\rho^n_{i+1}+\rho^n_i\right),\\
\left[F^n_{i+1/2}\right]_{preD} &= \gamma_p \Delta x\left(\rho^n_{i+1}-\rho^n_i\right),\\
\left[F^n_{i+1/2}\right]_D &= \nu_p \Delta x\left(\rho^n_{i+1}-\rho^n_i\right) \text{ and}\\
\left[F^{n+1/2}_{i+1/2}\right]_A &= \text{Limit}\left[\mu_p \Delta x\left(\left[\rho^{n+1/2}_{i+1}\right]_{FTCSD}-\left[\rho^{n+1/2}_i\right]_{FTCSD}\right)\right]
\end{aligned}
\tag{B.21}
$$

(this is the most general 1-D form of the algorithm: u is permitted to vary in time and space) and for the corrector step,

$$
\rho^{n+1}_i = \left[\rho^{n+1}_i\right]_{LD} + \frac{\left[F^n_{i+1/2}\right]_D - \left[F^n_{i-1/2}\right]_D}{\Delta x} - \frac{\left[F^{n+1}_{i+1/2}\right]_A - \left[F^{n+1}_{i-1/2}\right]_A}{\Delta x} \tag{B.22}
$$

where

$$
\left[\rho^{n+1}_i\right]_{LD} = \rho^n_i - \frac{\left[F^{n+1/2}_{i+1/2}\right]_T - \left[F^{n+1/2}_{i-1/2}\right]_T}{\Delta x} + \frac{\left[F^{n+1/2}_{i+1/2}\right]_{preD} - \left[F^{n+1/2}_{i-1/2}\right]_{preD}}{\Delta x} \tag{B.23}
$$

with the fluxes

$$
\begin{aligned}
\left[F^{n+1/2}_{i+1/2}\right]_T &= \frac{1}{2}u^{n+1/2}_{i+1/2}\Delta t\left(\rho^{n+1/2}_{i+1}+\rho^{n+1/2}_i\right),\\
\left[F^{n+1/2}_{i+1/2}\right]_{preD} &= \gamma_c \Delta x\left(\rho^{n+1/2}_{i+1}-\rho^{n+1/2}_i\right),\\
\left[F^n_{i+1/2}\right]_D &= \nu_c \Delta x\left(\rho^n_{i+1}-\rho^n_i\right) \text{ and}\\
\left[F^{n+1}_{i+1/2}\right]_A &= \text{Limit}\left[\mu_c \Delta x\left(\left[\rho^{n+1}_{i+1}\right]_{LD}-\left[\rho^{n+1}_i\right]_{LD}\right)\right].
\end{aligned}
\tag{B.24}
$$

The intermediate solution $\left[\rho^{n+1}_i\right]_{LD}$ uses time-centered fluxes in a diffused leapfrog (LD) difference scheme to make the algorithm second order in time. The LD algorithm plays the same role in the corrector step as the FTCSD algorithm does in the predictor step. Note that setting $\nu_c = \nu_p$, $\mu_c = \mu_p$ and $\gamma_c = \gamma_p = 0$ reduces (B.19)-(B.24) to those composing Boris's original ETBFCT algorithm.

Performing a Von Neumann analysis on the corrector step of the predictor-corrector formulation of Book and Fry's extended ETBFCT algorithm (this time with u constant to ease the analysis) gives an amplitude error

$$\begin{aligned} A = {} & \left[2(\mu_c - \nu_c - \gamma_c)\right]\beta^2 \\ & + \left[(\mu_c - \nu_c - \gamma_c)\left(\mu_c - \nu_c - \gamma_c - \frac{1}{6}\right) - 2\gamma_c(\mu_p - \nu_p - \gamma_p) - 2\gamma_c\mu_c \right. \\ & \left. + (\mu_p - 2(\nu_p + \gamma_p) + \nu_c)\varepsilon^2 + \frac{1}{4}\varepsilon^4\right]\beta^4 \\ & + O(\beta^6) \end{aligned} \tag{B.25}$$

and a phase error

$$\begin{aligned} R = {} & \left[-\frac{1}{6} + \mu_p - \nu_p - \gamma_p + \nu_c + \frac{1}{2}\gamma_c + \frac{1}{6}\varepsilon^2\right]\beta^2 \\ & + \left[\frac{1}{120} + \mu_p(\nu_p - \mu_p) - \nu_c(\mu_p + \mu_c) - \gamma_c(\nu_p + \gamma_p)\right. \\ & + \left(\mu_p - \nu_p - \gamma_p + \nu_c + \frac{1}{2}\gamma_c\right)\left(-\frac{1}{4} + \mu_p + \nu_c + \gamma_c\right) \\ & \left. + \frac{1}{2}\left(-\frac{1}{6} + \nu_p + \gamma_p + \frac{1}{2}\gamma_c\right)\varepsilon^2 - \frac{1}{20}\varepsilon^4\right]\beta^4 \\ & + O(\beta^6). \end{aligned} \tag{B.26}$$

To ensure good amplitude and phase accuracy in the full predictor-corrector formulation, the predictor and corrector step errors should be minimized individually. Thus, each set of coefficients chosen must simultaneously reduce the order of the amplitude and phase error in (B.14) and (B.15) (the predictor step error) as well as in (B.25) and (B.26) (the corrector step error). An extensive investigation of potential coefficient sets has produced several such sets, only two of which have proven to be useful in actual simulations.

The first set of coefficients uses those given in (B.17), Book and Fry's set, for the algorithm's predictor step and those in (B.18), with $\gamma_c = 0$, for its corrector step. (Note that the predictor step coefficients always have $\varepsilon = u(\Delta t/2)/\Delta x$ since the predictor step is a half timestep.) This set (Set II in Table B.1) makes the amplitude error fourth order in the predictor and sixth order in the corrector while the phase error becomes sixth order in the predictor and fourth order in the corrector. Unlike Book and Fry's uncorrected single timestep scheme, the amplitude error does not persist from timestep to timestep (Figure B.3) because the results of the predictor step, which suffer from significant attenuation at short wavelengths, are used only to calculate the fluxes required by the corrector step. The amplitude error of the corrector step is very small for low Courant numbers (there is none at all for $\varepsilon = 0$). The phase accuracy of the predictor step is very good and likely serves to increase the time-centeredness of the whole algorithm. The corrector step's phase accuracy, however, is merely adequate: any short wavelength modes that manage to develop lag their true positions and persist from timestep to timestep because of the algorithm's good corrector step amplitude accuracy. To be of practical use, this set of coefficients would need to be somehow modified to filter at short wavelengths so as to avoid the introduction of such modes and the subsequent generation of terraces.

The second set of coefficients found for the predictor-corrector formulation of Book and Fry's extended ETBFCT algorithm manages to address the shortcomings of the first by introducing a tunable element to the set. By requiring the predictor step to be at least fourth order accurate and the corrector step sixth order accurate in both amplitude and phase, the coefficients for the predictor step become

$$\nu_p = \frac{11}{30} + \frac{1}{30}\varepsilon^2,\ \mu_p = \frac{1}{6} - \frac{1}{6}\varepsilon^2 \text{ and } \gamma_p = -\frac{1}{5} + \frac{3}{10}\varepsilon^2 \text{ with } \varepsilon = u\frac{\Delta t}{2\Delta x}, \quad \text{(B.27)}$$

and for the corrector step,

Coefficients	ν_p	μ_p	γ_p	ν_c	μ_c	γ_c
Boris's original set (Set I)	$\frac{1}{6}+\frac{1}{3}\varepsilon^2$	$\frac{1}{6}-\frac{1}{6}\varepsilon^2$	0	$\frac{1}{6}+\frac{1}{3}\varepsilon^2$	$\frac{1}{6}-\frac{1}{6}\varepsilon^2$	0
Book and Fry's extended set (Set II)	$-\frac{1}{30}+\frac{2}{15}\varepsilon^2$	$\frac{1}{6}-\frac{1}{6}\varepsilon^2$	$\frac{1}{5}+\frac{1}{5}\varepsilon^2$	$\frac{1}{6}-\frac{1}{24}\varepsilon^2$	$\frac{1}{6}-\frac{1}{24}\varepsilon^2$	0
This paper's set (Set III)	variable	$\frac{1}{6}-\frac{1}{6}\varepsilon^2$	$\frac{1}{6}+\frac{1}{3}\varepsilon^2-\nu_p$	$\frac{1}{6}+\frac{1}{48}\varepsilon^2-\frac{1}{240}K$	$\frac{1}{6}-\frac{5}{48}\varepsilon^2+\frac{1}{240}K$	$-\frac{1}{8}\varepsilon^2+\frac{1}{120}K$

Table B.1. Diffusion, antidiffusion and prediffusion coefficient sets for the predictor-corrector formulation of Book and Fry's extension to ETBFCT given in (B.19)-(B.24). The subscripts p and c indicate the predictor and corrector step coefficients, respectively. $\varepsilon = u(\Delta t/2)/\Delta x$ for the predictor step coefficients and $\varepsilon = u\Delta t/\Delta x$ for the corrector step coefficients with $K = \left[5\left(41\varepsilon^4 + 160\left(3\nu_p - 1\right)\left(\varepsilon^2 - 4\right) + 64\right)\right]^{1/2}$.

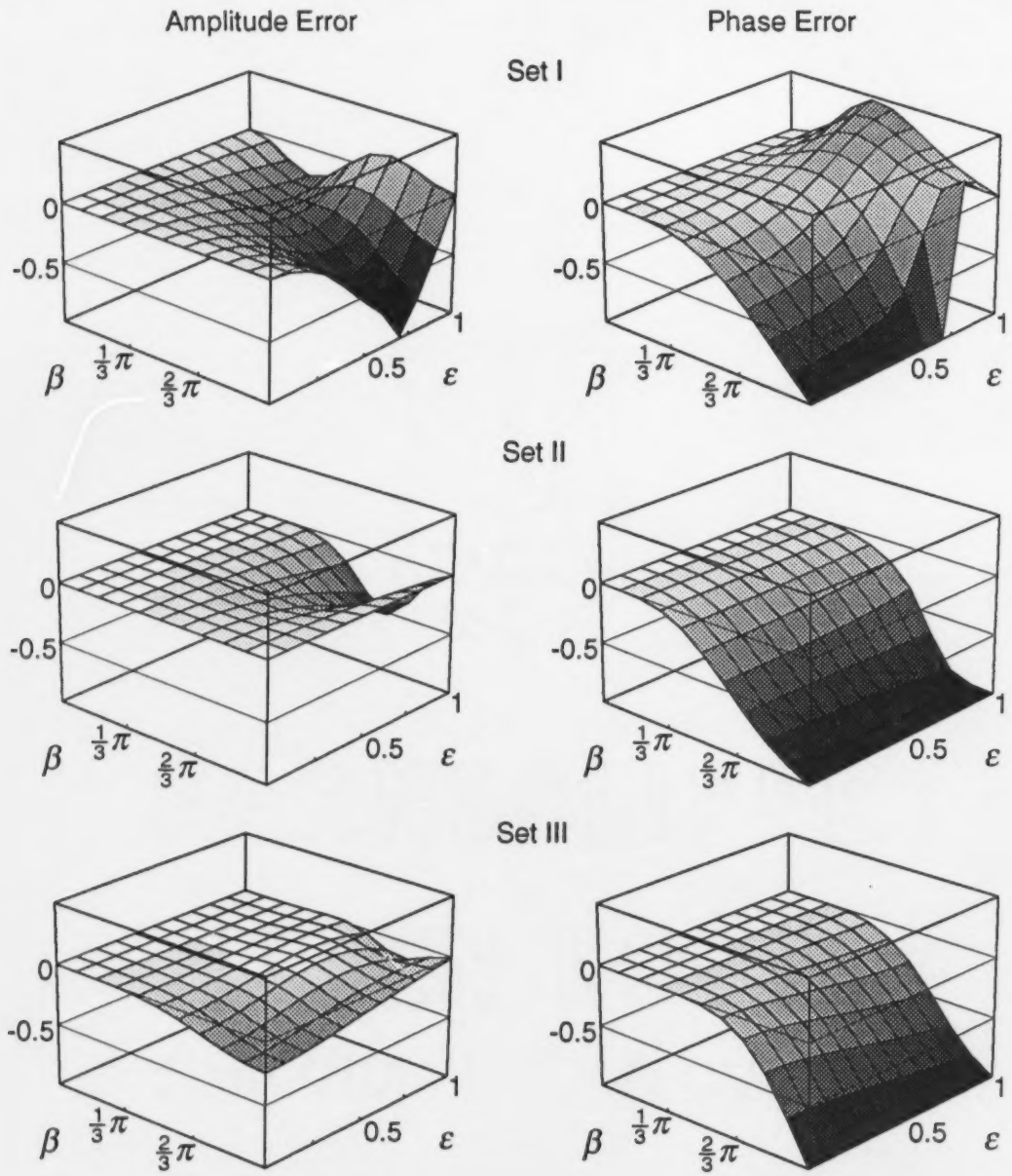


Figure B.3. The exact amplitude and phase error of the corrector step of the predictor-corrector ETBFCT algorithm for three sets of coefficients: Set I is Boris's original set, Set II is Book and Fry's extended phase accurate set and Set III (with $\nu_p = 0.36467$) is this appendix's tunable set (see Table B.1).

$$\nu_c = \frac{1}{6} - \frac{1}{24}\varepsilon^2,\ \mu_c = \frac{1}{6} - \frac{1}{24}\varepsilon^2 \text{ and } \gamma_c = 0 \text{ with } \varepsilon = u\frac{\Delta t}{\Delta x}. \qquad \text{(B.28)}$$

These coefficients make the algorithm unstable due to residual amplitude errors in both the predictor and corrector steps. However, leaving ν_p as a free parameter and removing the requirement of sixth order amplitude accuracy in the corrector step gives the algorithm an element of tunability while maintaining at least fourth order amplitude accuracy in the corrector step. The new coefficients (Set III in Table B.1) are

$$\mu_p = \frac{1}{6} - \frac{1}{6}\varepsilon^2,\ \gamma_p = \frac{1}{6} + \frac{1}{3}\varepsilon^2 - \nu_p \text{ with } \varepsilon = u\frac{\Delta t}{2\Delta x} \qquad \text{(B.29)}$$

and

$$\nu_c = \frac{1}{6} + \frac{1}{48}\varepsilon^2 - \frac{1}{240}K,\ \mu_c = \frac{1}{6} - \frac{5}{48}\varepsilon^2 + \frac{1}{240}K,\ \gamma_c = -\frac{1}{8}\varepsilon^2 + \frac{1}{120}K,$$
$$K = \left[5\left(41\varepsilon^4 + 160\left(3\nu_p - 1\right)\left(\varepsilon^2 - 4\right) + 64\right)\right]^{1/2} \text{ with } \varepsilon = u\frac{\Delta t}{\Delta x}. \qquad \text{(B.30)}$$

Adjusting ν_p appropriately and substituting into (B.29) and (B.30) results in sixth order phase accuracy during the corrector step and an adjustable amount of filtering at short wavelengths which can be used to control the formation of terraces (see Figures B.2 and B.3). Filtering at short wavelengths and low phase error are both essential in inhibiting the formation of terraces.

TEST PROBLEMS

To evaluate the performance of the predictor-corrector formulation of Book and Fry's extended ETBFCT algorithm for the different coefficient sets, I subjected an implementation of the algorithm to three test problems. The first problem, the semicircle test, examines the algorithm's tendency towards phase, amplitude and terracing errors in one dimension [*Oran and Boris*, 1987]. The

second, the shock tube test, determines the algorithm's suitability for one-dimensional problems involving shocks, discontinuities and rarefactions [*Hirsch*, 1988; *Sod*, 1978]. The third, the Kelvin-Helmholtz (KH) instability test, verifies an algorithm's ability to deal with two-dimensional hydrodynamic systems having a wide range of convection velocities. This last test problem is much more involved than the first two and has no analytic solution, but can point out weaknesses in the algorithm that are missed by the other two tests.

In the semicircle test problem a semicircular density profile is convected at a uniform rate across a periodic simulation grid by repeated solution of (B.1). The density is initially uniform everywhere except between cells 2 and 32 of the 80 cell simulation grid where it assumes a semicircular profile. The results of the test are evaluated by comparing the algorithm-convected density profile against the exact solution.

The shock tube test problem applies the algorithm to the solution of the one dimensional hydrodynamic equations,

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} &= 0 \\ \frac{\partial (\rho u)}{\partial t} + \frac{\partial \left(\rho u^2\right)}{\partial x} &= -\frac{\partial p}{\partial x} \\ \frac{\partial E}{\partial t} + \frac{\partial (Eu)}{\partial x} &= -\frac{\partial (\rho u)}{\partial x}, \end{aligned} \tag{B.31}$$

where p is pressure, $E = p/(\gamma - 1) + \rho u^2/2$ and $\gamma = 1.4$, for a rapidly evolving one dimensional physical system. The physical system modelled in the problem consists of two adjacent regions of stationary gas with different densities and pressures initially separated by a diaphragm. To the left of the diaphragm, $\rho = 1$ and $p = 10^5$ while to the right, $\rho = 0.1$ and $p = 10^4$ (values are in SI units). The test starts with the disintegration of the diaphragm and follows the evolution of the flows that develop. The simulation grid used in this test is 80 cells long with

$\Delta x = 0.125$ and fixed boundary conditions. An exact analytic solution is possible for this situation [*Hirsch*, 1988] and permits the results of the algorithm to be objectively evaluated.

The KH instability test problem applies the algorithm to the two dimensional set of hydrodynamic equations to examine how well it can model a slowly evolving two dimensional fluid system. The test simulates the evolution of an unstable flow consisting of two adjacent infinite streams with different densities and antiparallel velocities which are subjected to a small perturbation. For appropriate initial conditions, the perturbation will grow exponentially at a rate predicted by a stability analysis of the hydrodynamic equations [*Drazin and Reid*, 1981; *Miura and Pritchett*, 1982]: this is the KH instability. The KH instability is a particularly good test of the convective properties of the algorithm because the component of velocity perpendicular to the flow in the infinite streams is initially very small and increases exponentially as the instability develops until it eventually becomes comparable with the fastest convection speeds in the flow, thus effectively scanning the algorithm for its performance over a wide range of convection velocities. Even in its saturated, nonlinear stage, the KH instability still has a wide range of convection speeds and scale sizes which continue to tax the algorithm.

The KH instability simulation uses a 48×46 ($N_x \times N_y$) grid with $\Delta x = 3.162\times10^{-4}$ and $\Delta y = 3.130\times10^{-4}$ which has periodic boundaries at $x = 0$ and $x = 0.0152$ and slip wall boundaries at $y = \pm7.2\times10^{-3}$. The problem is initialized with $p = 1.013\times10^{5}$ and with $\rho = 2$ and 1 in the upper and lower streams, respectively. The velocity jump across the shear layer is equal to the sound speed in the lower stream, $\Delta v = 410.9$, and the velocities of each stream are chosen so as to put the instability into a stationary reference frame. The density and velocity vary across the grid according to a hyperbolic tangent function centered at $y = 0$ with half-width $a = 0.001$. This unstable equilibrium is then perturbed by an eigenmode solution of the linearized hydrodynamic

equations [*Miura and Pritchett*, 1982] for the longest KH wave mode that will fit the periodic simulation grid. Doing so ensures that no other mode will be triggered into growth during the linear phase of the instability and reduces the possibility that the small scale structures that sometimes appear in the nonlinear phase of the test are due to physical processes rather than problems with the algorithm.

The algorithm used in the tests is the predictor-corrector formulation of Book and Fry's extension to ETBFCT given in (B.19)–(B.24) with the Limit[ ] operation replaced by a call to the FLIMIT subroutine [*Zalesak*, 1979; *Book et al.*, 1981]. For the semicircle and shock tube tests, the logical variables JPRLIM, PRLIM and FOLD passed to FLIMIT are all set false while for the KH instability test problem, JPRLIM is set false and PRLIM and FOLD are set true. The latter problem is a two dimensional one which requires that the algorithm be extended to two dimensions: the FLIMIT subroutine makes such an extension relatively straightforward. Note that there is an error in the listing of FLIMIT given in *Book et al.* [1981]: line 240 should be moved to follow line 243.

TEST RESULTS AND ANALYSIS

The results of the semicircle test problem (Figures B.4 and B.5) confirm the amplitude and dispersion properties of the algorithm previously determined by the Von Neumann analysis (Figure B.3) and differ significantly from the results of the single step algorithm (see Figure B.1). Comparison of Figure B.3 with Figure B.2 shows that the favorable amplitude and phase properties seen for the predictor step when using the original set of coefficients (Set I in Table B.1) have been lost with the addition of the corrector step. A region of positive amplitude error has appeared at high Courant numbers in the corrector step making the algorithm unstable for $\varepsilon > 0.9$ where it was stable in the predictor step. A large region of strong attenuation affecting fast moving modes has also appeared and is responsible for the diffusive behavior seen for $\varepsilon = 0.9$ and 0.25 in Figure B.4.

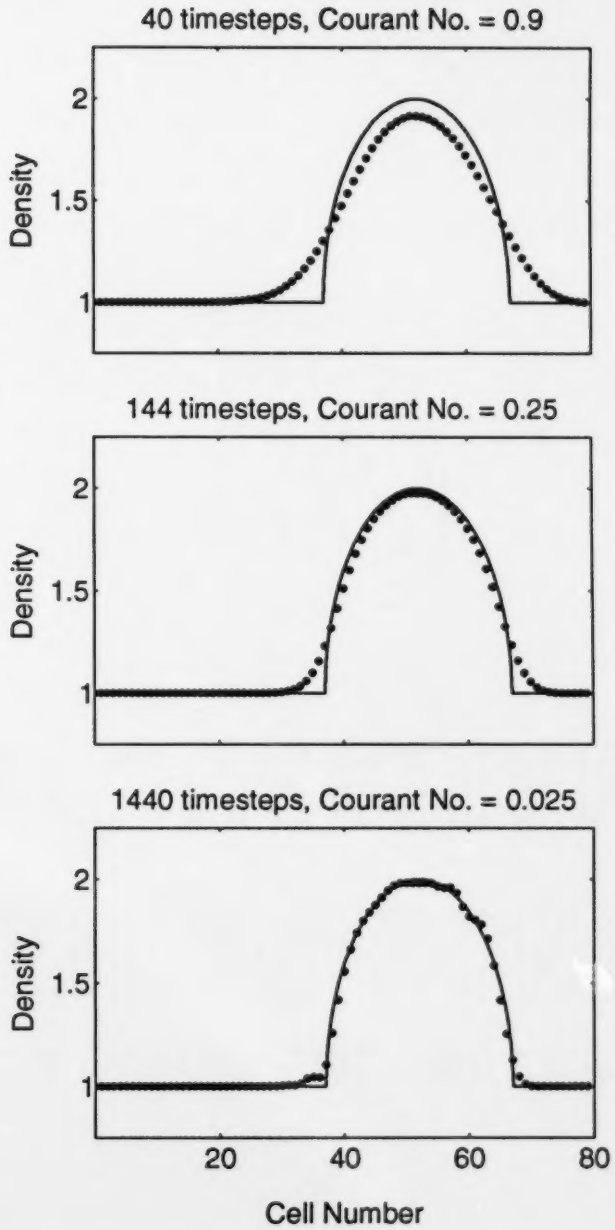


Figure B.4. A semicircular density profile after being convected 36 cells by the predictor-corrector form of the original ETBFCT algorithm for a range of Courant numbers.

This sizable amplitude error masks the effects of the positive phase error by eroding away those modes that would be most affected by the phase error before the error can manifest itself. As the Courant number decreases further, the amplitude error reduces to the point where the predictor-corrector form of the algorithm produces results that are nearly identical to those of its single step formulation (compare Figures B.1 and B.4).

The set of coefficients introduced in (B.29) and (B.30) (Set III in Table B.1), however, give the opposite behavior in the semicircle convection test for appropriately chosen ν_p: rather than increasing with decreasing timestep size, the phase and terracing errors decrease (see Figure B.5). Once again, the plots of amplitude and phase error show why this is so (see Figure B.3). Setting $\nu_p = 0.36467$ in this coefficient set adds attenuation to the algorithm which serves to filter out short wavelength modes before the dispersion error experienced by those modes becomes significant. The effect of filtering is strongest for slow moving modes and makes the results of the semicircle test at low Courant numbers appear more diffuse than for high values. For this particular value of ν_p, the highest Courant number for which the algorithm is guaranteed to be stable is ~0.4. In spite of this low value, the algorithm still produces results for $\varepsilon = 0.9$ which look as good as those produced for the Set I coefficients when $\varepsilon = 0.025$.

The Set III coefficients can give surprisingly bad predictor step performance (see Figure B.2). The amplitude error of the algorithm for these coefficients, with $\nu_p = 0.36467$, makes the predictor step unstable for all but the longest wavelength modes at most Courant numbers and gives the worst phase error of the three coefficient sets in Table B.1. Fortunately, the predictor step results are used only to determine values for the corrector step fluxes so the errors introduced during the predictor step do not persist from timestep to timestep as directly as they would for a single step algorithm. However, the errors introduced by the predictor step lessen the accuracy of the time advance since the predictor

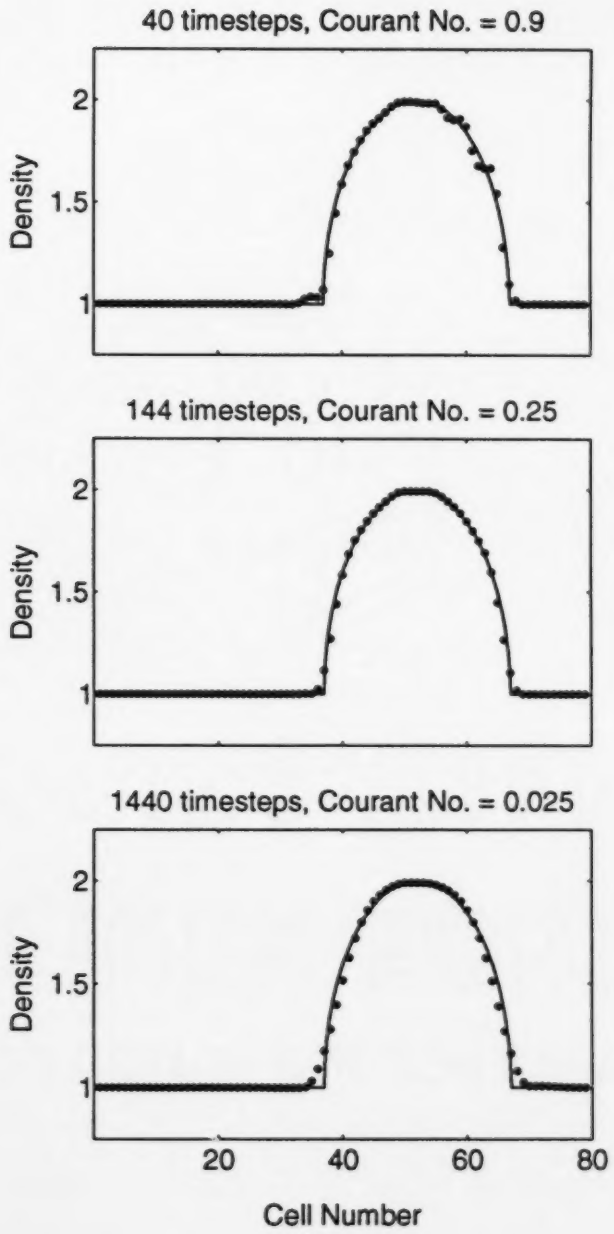


Figure B.5. A semicircular density profile after being convected 36 cells by the predictor-corrector form of Book and Fry's extended ETBFCT algorithm with this appendix's tunable coefficients (Set III in Table B.1 with $\nu_p = 0.36467$) for a range of Courant numbers.

step exists primarily to make the difference scheme second order accurate in time through a time-centered difference. Decreasing the size of the timestep should help the algorithm recover from the errors created by the predictor step without leading to the terracing problems that arise for small Courant numbers when using the Set I coefficients.

The results of the shock tube test presented in Figures B.6 and B.7 further confirm the performance characteristics of the Set I and III coefficients. The density profile for the Set I coefficients is completely free from terracing errors because the choice of $\varepsilon = 0.9$ for the simulation and the existence of high convection velocities places the simulation in the trough of the amplitude error shown in Figure B.3. Any phase errors that develop are immediately attenuated, thus accounting for the smoothness of the expansion fan extending from $x = 2.57$ to 4.85. Figure B.7 shows that the reduced attenuation afforded by the Set III coefficients (with $\nu_p = 0.36467$) is not strong enough to keep the phase error from producing terraces in the expansion fan. The algorithm's improved amplitude accuracy also appears to be responsible for keeping the contact discontinuity at $x = 6.90$ quite steep. Interestingly, the region of positive amplitude error at intermediate wavelengths and high Courant numbers for the Set III coefficients in Figure B.3 has not introduced any significant errors into the solution, with a possible exception being the leading edge of the shock ($x = 8.53$). This unexpected stability is probably a result of leakage from the unstable modes into adjacent ones where there is attenuation.

The results of the KH instability test problem shown in Figure B.8 illustrate the sensitivity of this test to terracing effects. When Boris's original set of coefficients (Set I) are used in the algorithm, the vortical structure that appears in the nonlinear stage of the KH instability is beset by short-scale disturbances. The disturbances originate near the stagnation point in the "braid" region of the wave and have the effect of roughening the interface between the light and dense streams. This roughened appearance is similar to the staircase appearance of

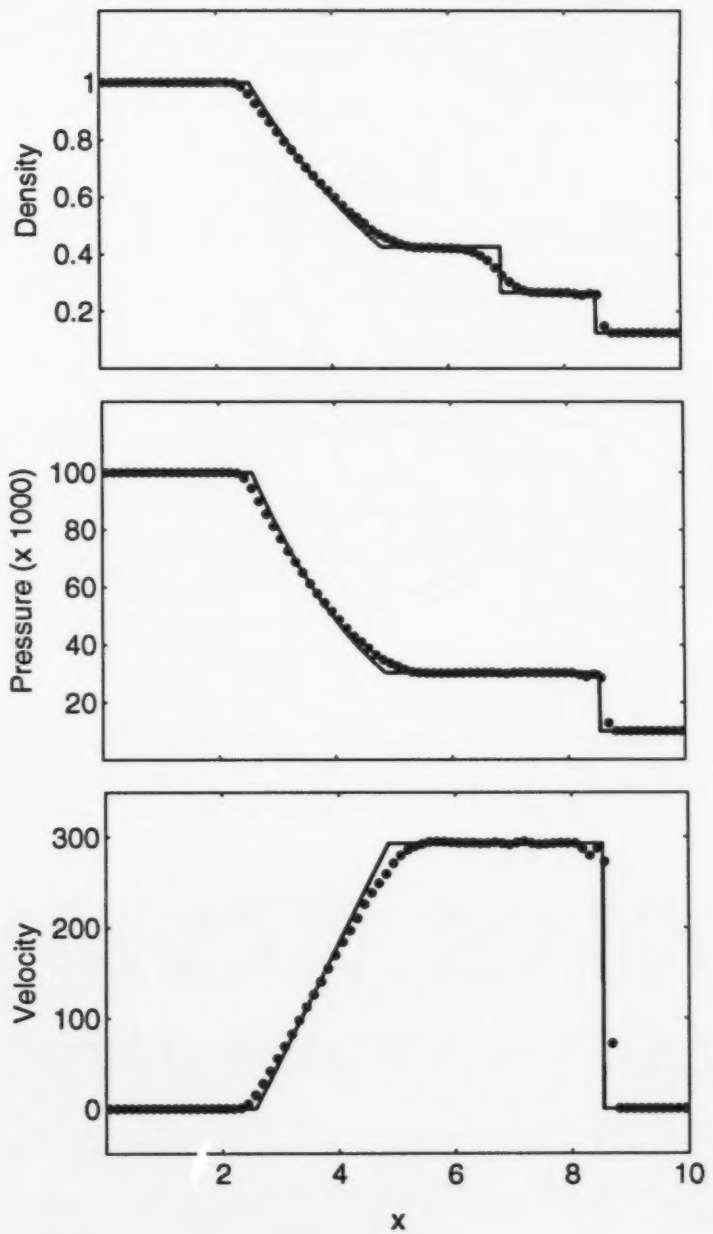


Figure B.6. The one dimensional shock tube at $t = 6.496 \times 10^{-3}$ for the predictor-corrector form of the original ETBFCT algorithm. (Courant number = 0.9)

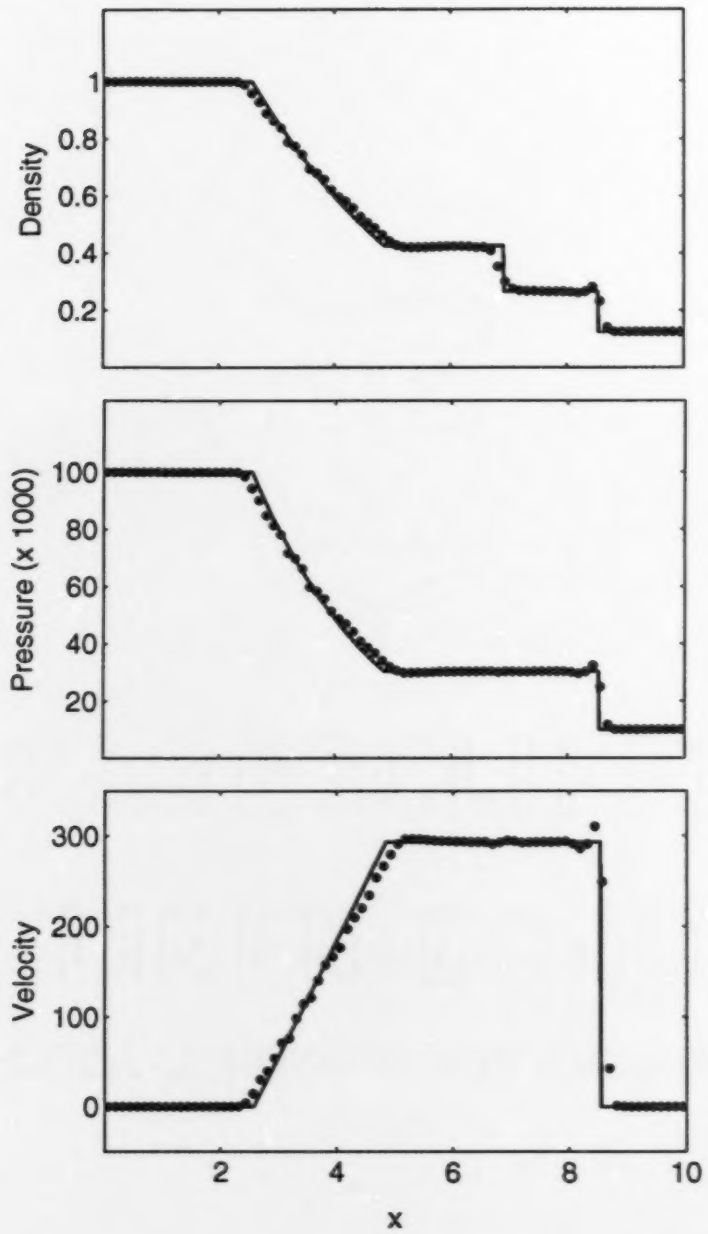


Figure B.7. The one dimensional shock tube at $t = 6.496 \times 10^{-3}$ for the predictor-corrector form of Book and Fry's extended ETBFCT algorithm with this appendix's tunable coefficients (Set III in Table B.1 with $\nu_p = 0.36467$). (Courant number = 0.9)

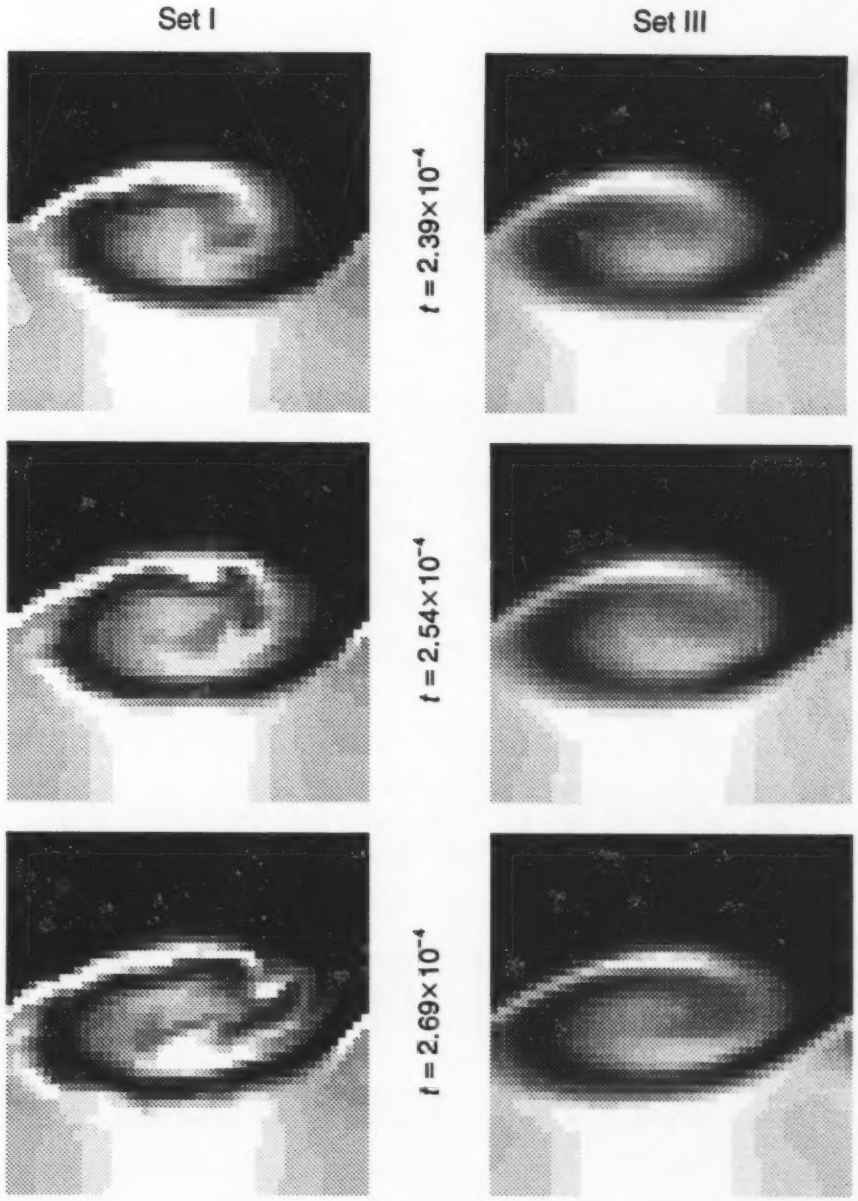


Figure B.8. Density in the nonlinear stage of the KH instability using Boris's coefficient set (Set I in Table B.1) and this appendix's tunable set (Set III in Table B.1 with $\nu_p = 0.36617$). White indicates $\rho \leq 1$ while black indicates $\rho \geq 2$. The "billow" is the spiral-shaped region and the "braid" is the straight section connecting billows. (Courant number = 0.25)

shocks generated during *Woodward and Colella*'s [1984] test of the original ETBFCT algorithm using a simulated supersonic wind tunnel. In both cases, the structures on which the disturbances appear move very slowly relative to the simulation grid and so would be expected to exhibit phase errors for the Set I coefficients.

In the KH instability test, the small bumps that appear on the density interface for the Set I coefficients are caught by the main flow and carried away from the stagnation point and into regions of strong vorticity where they act as seeds for the development of a secondary KH instability. The secondary instability grows as the disturbances convect into the "billow" region and toward the center of rotation. Soon the secondary instability is large enough to disturb the braid region where it was initiated and begins a feedback process which breaks the large organized vortical flow into many small vortices.

This appendix's coefficient set (Set III in Table B.1) can be tuned to minimize the dispersion error that causes the algorithm to fail for the original ETBFCT coefficients (Set I). The value of ν_p used in the previous two test problems filters too strongly in this case: the shear layer broadens because of excessive filtering and loses the intermediate scale structures that are necessary to the development of the nonlinear phase of the instability. A slightly larger value of ν_p, $\nu_p = 0.36617$, reduces the filtering to a level that permits the nonlinear phase to develop and prevents terraces from forming, but still causes premature broadening of the shear layer. To avoid artificially widening the shear layer, the simulation uses Book and Fry's (Set II) coefficients during the linear stage of the KH instability's development when the gradients have little short wavelength content. Those coefficients do not filter stationary structures, such as the cross-stream density profile, but do suffer from phase errors at short wavelengths. Once the instability has evolved to the point where gradients begin to steepen, and thus generate short wavelength modes, the simulation switches over to the Set III coefficients (with $\nu_p = 0.36617$) to ensure that no such modes can persist

long enough to cause the problems that lead to the secondary KH instability.

DISCUSSION AND CONCLUSIONS

The predictor-corrector formulation of the extended ETBFCT algorithm presented by Book and Fry behaves rather differently for the new coefficients presented in this appendix (Set III in Table B.1) when compared with Boris's original coefficients (Set I in Table B.1). For the Set I coefficients, the algorithm has been shown to become increasingly sensitive to dispersion errors which lead to terracing as the Courant number decreases, while for the Set III coefficients with appropriately chosen v_p, the algorithm becomes less susceptible to terracing with decreasing Courant number (see Figure B.4 and B.5). Both types of behavior are desirable, but for complementary physical problems. For problems where structures move quickly across a simulation grid, such as the blast problems for which ETBFCT was originally devised, Boris's coefficient set is to be preferred because the strong damping at high Courant numbers it affords prevents terracing errors from developing. In fact, there is also an element of tunability present for Boris's coefficients: if the structures seen in a high Courant number simulation appear too diffuse, reducing the Courant number will reduce the attenuation that causes the diffusion and make those features sharper. For problems which contain a wide range of convection velocities and require slow moving structures to be accurately convected, the new coefficient set is preferable. By adjusting the Courant number and tuning the coefficient v_p, it should be possible to achieve good performance for a wide range of fluid systems using the new coefficients.

The KH instability test problem shows that there are limits to the abilities of the new coefficients. To pass this test, the simulation used Book and Fry's amplitude accurate coefficients (Set II in Table B.1) during the linear phase of the instability to avoid over-filtering the stationary structures crucial to the development of the instability and then had to switch to the new coefficients

during the nonlinear phase to filter away potential dispersion errors which would otherwise disturb the flow. It seems preferable for the algorithm to use a single set of coefficients which are set once at the beginning of a simulation and never changed[4].

Although it does not appear possible to directly combine the excellent predictor step properties of the Set II coefficients with the superior corrector step properties of the Set III coefficients into a single set of coefficients, it may be possible to achieve similar results by blending the two sets together. By switching between the two coefficient sets on alternate timesteps, a simulation might derive the benefits of both sets of coefficients. Preliminary tests of such a scheme show that the strength of the filter can be decreased in this way without producing too much more dispersion. Further experiments suggest that an adaptive blending of the two sets of coefficients in response to the presence of short wavelength and potentially dispersive modes in the simulation will further improve the dispersion and amplitude accuracy of the algorithm.

The efforts made here to reduce dispersion in the ETBFCT algorithm are also useful in developing and improving the large eddy simulation (LES) properties of the algorithm. Boris has argued [*Boris*, 1990] that the FCT technique has a minimal LES filter and matched subgrid model already built-in. The modifications to the ETBFCT algorithm presented here alter that filter to ensure that the short wavelength modes that are most susceptible to dispersion are filtered out. Those same modifications make the filter less dependent on the convection velocity (compare the amplitude errors for the Set I and Set III coefficients given in Figure B.3), a desirable attribute for a practical LES algorithm, and permit the filter to be adjusted to give the best possible LES properties.

The advantages of the new set of ETBFCT coefficients (Set III in Table B.1) presented in this appendix are many. Using the new coefficients in the extended

---

[4] I have developed such a set of coefficients since this paper was submitted. The new coefficients are used in the simulations presented in the body of this thesis and in Appendix C.

time-centered ETBFCT algorithm makes the solution to the continuity equation always sixth order phase accurate and permits the algorithm to be tuned to give an adjustable amount of short wavelength filtering while maintaining at least fourth order amplitude accuracy. Tunable filtering allows the algorithm to be adjusted to eliminate terracing in simple convection problems as well as demanding two dimensional hydrodynamic simulations. The performance of the algorithm becomes more predictable (i.e., terracing and phase errors become less) as the Courant number decreases thus increasing the algorithm's reliability in slow flow simulations. Implementation of the new coefficients requires very little modification of the existing ETBFCT algorithm and yet produces significant improvements in the algorithm.

APPENDIX C

THREE-DIMENSIONAL IDEAL MHD SIMULATION

The following is a slightly simplified version of the code I developed to perform the simulations presented in the paper and discussed in Appendix B. As it is currently configured, the program simulates the development of a localized pressure perturbation in an otherwise uniform plasma using a mesh. The final values of the physical variables in a small, two-dimensional subsection of the simulation space immediately follow the program listing.

Computational resources required:

3.2 MBytes of program memory

13.5 minutes of CPU time (when run on an IBM RS/6000 PowerStation 320)

```
      PROGRAM IMHD3D

C     Simulate the evolution of an ideal, MHD fluid in three dimensions.

C     The program must be compiled by a FORTRAN 77 compiler.
C     The program was developed on DEC Vax and IBM RS/6000 computers
C     and makes use of some of the extensions available on those machines:
C     it will likely require some modification before it will run on
C     other computers.

C     Created by:
C       John R. Manuel
C       Department of Physics / Canadian Network for Space Research
C       University of Alberta
C       Edmonton, Alberta
C       T6G 2J1 CANADA
C       Phone:              (403) 492-2526
C       E-mail (Internet):  manuel@space.ualberta.ca

      INTEGER IT, ITMAX
      REAL*8 CFL, DT, TIME, TMAX
      COMMON /TIME/   DT, IT, ITMAX, TIME, TMAX, CFL
      INTEGER NXM1, NYM1, NZM1, OUT_NUM
      REAL*8 IDX, IDY, IDZ, IVOL, I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2
      CHARACTER EQN*3, ID*8
      COMMON /MISC/   OUT_NUM, NXM1, NYM1, NZM1, IDX, IDY, IDZ, IVOL,
```

```
     2              I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2,
     3              ID, EQN(8)
      REAL*8 OUTDT, OUTSTART
      DATA OUTDT      / 0.1D0 /
      DATA OUTSTART   / 0.D0 /

D     PRINT *, '-> DEBUGGING LINES ACTIVE <-'

C     Make the grid and calculate some grid-dependent quantities...
      CALL MKGRID

C     Initialize the physical variables...
      CALL INITIAL

C     Set the initial timestep size...
      CALL SETDT

C     Write out the initialization...
      CALL DATA_OUT

C     The main loop...
      DO IT = 1, ITMAX
        TIME = TIME + DT
C       Time-advance the physics to step IT and time TIME...
        CALL ADVANCE_ALL
C       Set the timestep size every now and then...
        IF (MOD(IT, 10) .EQ. 0)  CALL SETDT
        IF (TIME .GE. OUTSTART
     2      .AND.
     3      MOD(TIME, OUTDT) .LT. DT) THEN
C         Dump physical variables at regular time intervals...
          CALL DATA_OUT
        ELSE IF (TIME .GE. TMAX) THEN
          CALL DATA_OUT
C         Quit when time limit exceeded...
          PRINT *, ' TIME is', TIME
          PRINT *, ' TMAX exceeded.'
          CALL FINISH
        END IF
      END DO
      CALL DATA_OUT
C     Quit when iteration count exceeded...
      PRINT *, ' TIME is', TIME
      PRINT *, ' ITMAX exceeded.'
      CALL FINISH

      E N D


      SUBROUTINE MKGRID

C     Initialize the computational grid. X(I), Y(J), Z(K), and VOL
C     give the physical coordinates and volume of a cell centred on (I,J,K).
```

```
      INTEGER NNX, NNY, NNZ
      PARAMETER (NNX = 22, NNY = 22, NNZ = 22)
      INTEGER NX, NY, NZ
      REAL*8 X, Y, Z, DX, DY, DZ, VOL
      COMMON /GRID/   X(NNX), Y(NNY), Z(NNZ),
     2                DX, DY, DZ, VOL, NX, NY, NZ
      INTEGER NXM1, NYM1, NZM1, OUT_NUM
      REAL*8 IDX, IDY, IDZ, IVOL, I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2
      CHARACTER EQN*3, ID*8
      COMMON /MISC/   OUT_NUM, NXM1, NYM1, NZM1, IDX, IDY, IDZ, IVOL,
     2                I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2,
     3                ID, EQN(8)
      INTEGER I
      REAL*8 OFFSET

D     PRINT *, '-> MKGRID'

      OFFSET = -1.5D0*DX
      DO I = 1, NX
        X(I) = DX*I + OFFSET
      END DO

      OFFSET = -1.5D0*DY
      DO I = 1, NY
        Y(I) = DY*I + OFFSET
      END DO

      OFFSET = -1.5D0*DZ
      DO I = 1, NZ
        Z(I) = DZ*I + OFFSET
      END DO

      VOL = DX*DY*DZ

C     Define some grid-related and often-used constants...
      NXM1 = NX - 1
      NYM1 = NY - 1
      NZM1 = NZ - 1
      IDX = 1.D0 / DX
      IDY = 1.D0 / DY
      IDZ = 1.D0 / DZ
      IVOL = 1.D0 / VOL
      I2DX = 0.5D0*IDX
      I2DY = 0.5D0*IDY
      I2DZ = 0.5D0*IDZ
      IDX2 = IDX**2
      IDY2 = IDY**2
      IDZ2 = IDZ**2

D     PRINT *, '<- MKGRID'
      RETURN

      E N D
```

```
      SUBROUTINE INITIAL

C     Initialize the physical variables.

C     The present initialization sets up a localized pressure
C     perturbation in an otherwise uniform plasma.

      INTEGER NNX, NNY, NNZ
      PARAMETER (NNX = 22, NNY = 22, NNZ = 22)
      REAL*8 RHO, RVX, RVY, S, GAMMA, BX, BY, BZ, RVZ
      COMMON /PHYSA/  RHO(NNX,NNY,NNZ),
     2                RVX(NNX,NNY,NNZ), RVY(NNX,NNY,NNZ),
     3                S(NNX,NNY,NNZ), GAMMA
      COMMON /PHYSB/  BX(NNX,NNY,NNZ), BY(NNX,NNY,NNZ), BZ(NNX,NNY,NNZ)
      COMMON /PHYSC/  RVZ(NNX,NNY,NNZ)
      INTEGER NX, NY, NZ
      REAL*8 X, Y, Z, DX, DY, DZ, VOL
      COMMON /GRID/   X(NNX), Y(NNY), Z(NNZ),
     2                DX, DY, DZ, VOL, NX, NY, NZ
      INTEGER IT, ITMAX
      REAL*8 CFL, DT, TIME, TMAX
      COMMON /TIME/   DT, IT, ITMAX, TIME, TMAX, CFL
      REAL*8 BC
      COMMON /BC/     BC(6,8)
      INTEGER NXM1, NYM1, NZM1, OUT_NUM
      REAL*8 IDX, IDY, IDZ, IVOL, I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2
      CHARACTER EQN*3, ID*8
      COMMON /MISC/   OUT_NUM, NXM1, NYM1, NZM1, IDX, IDY, IDZ, IVOL,
     2                I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2,
     3                ID, EQN(8)
      INTEGER I, J, K, UNIT_LEN
      REAL*8 B0, CS, CA, CF, P0, RHO0, RHOPERT
      CHARACTER UNIT*80

D     PRINT *, '-> INITIAL'

C     Define the polytropic index...
      GAMMA = 5.D0 / 3.D0

      P0 = 1.D0
C     Choose RHO0 and B0 to make CF/CS/CA = 1.0/0.8/0.6
      RHO0 = GAMMA*P0/0.8D0**2
      B0 = 0.6D0*SQRT(RHO0)
      CS = SQRT(GAMMA*P0/RHO0)
      CA = SQRT(B0**2/RHO0)
      CF = SQRT(CS**2 + CA**2)
      DO K = 1, NZ
        DO J = 1, NY
          DO I = 1, NX
            RHOPERT = RHO0*0.1D0*EXP(-0.5D0*((DBLE(I) - 11.5D0)**2
     2                                     + (DBLE(J) - 11.5D0)**2
     3                                     + (DBLE(K) - 11.5D0)**2))
            RHO(I,J,K) = RHO0 + RHOPERT
```

```
            RVX(I,J,K) = 0.D0
            RVY(I,J,K) = 0.D0
            RVZ(I,J,K) = 0.D0
            S(I,J,K) =  P0 + CS**2*RHOPERT
            BX(I,J,K) = 0.D0
            BY(I,J,K) = 0.D0
            BZ(I,J,K) = B0
          END DO
        END DO
      END DO

C     Do boundary conditions...
      CALL BCOND (RHO, RHO, 1)
      CALL BCOND (RVX, RVX, 2)
      CALL BCOND (RVY, RVY, 3)
      CALL BCOND (RVZ, RVZ, 4)
      CALL BCOND (S,   S,   5)
      CALL BCOND (BX,  BX,  6)
      CALL BCOND (BY,  BY,  7)
      CALL BCOND (BZ,  BZ,  8)

C     File to receive formatted snapshots of array subsections...
      UNIT = 'array_subs.dat'
      UNIT_LEN = INDEX(UNIT, ' ') - 1
      IF (UNIT_LEN .EQ. -1)  UNIT_LEN = LEN(UNIT)
      OPEN (12, FILE = UNIT(1:UNIT_LEN),
     2      STATUS = 'NEW', FORM = 'FORMATTED')

      RETURN

      E N D


      SUBROUTINE SETDT

C     Calculate a timestep size that satisfies the Courant condition

      INTEGER NNX, NNY, NNZ
      PARAMETER (NNX = 22, NNY = 22, NNZ = 22)
      REAL*8 RHO, RVX, RVY, S, GAMMA, BX, BY, BZ, RVZ
      COMMON /PHYSA/  RHO(NNX,NNY,NNZ),
     2                RVX(NNX,NNY,NNZ), RVY(NNX,NNY,NNZ),
     3                S(NNX,NNY,NNZ), GAMMA
      COMMON /PHYSB/  BX(NNX,NNY,NNZ), BY(NNX,NNY,NNZ), BZ(NNX,NNY,NNZ)
      COMMON /PHYSC/  RVZ(NNX,NNY,NNZ)
      INTEGER NX, NY, NZ
      REAL*8 X, Y, Z, DX, DY, DZ, VOL
      COMMON /GRID/   X(NNX), Y(NNY), Z(NNZ),
     2                DX, DY, DZ, VOL, NX, NY, NZ
      INTEGER IT, ITMAX
      REAL*8 CFL, DT, TIME, TMAX
      COMMON /TIME/   DT, IT, ITMAX, TIME, TMAX, CFL
      REAL*8 BC
      COMMON /BC/     BC(6,8)
```

```
      INTEGER NXM1, NYM1, NZM1, OUT_NUM
      REAL*8 IDX, IDY, IDZ, IVOL, I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2
      CHARACTER EQN*3, ID*8
      COMMON /MISC/   OUT_NUM, NXM1, NYM1, NZM1, IDX, IDY, IDZ, IVOL,
     2                I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2,
     3                ID, EQN(8)
      INTEGER I, J, K
      REAL*8 CA2, CS2, CMAX, DTTEMP, DTTEMPMAX, IRHO, VX, VY, VZ

D     PRINT *, '-> SETDT'

C     Scan for the maximum velocity...
      DTTEMPMAX = 0.D0
      DO K = 2, NZM1
        DO J = 2, NYM1
          DO I = 2, NXM1
            IRHO = 1.D0/RHO(I,J,K)
            VX = ABS(IRHO*RVX(I,J,K))
            VY = ABS(IRHO*RVY(I,J,K))
            VZ = ABS(IRHO*RVZ(I,J,K))
            CS2 = GAMMA*S(I,J,K)*IRHO
            CA2 = (BX(I,J,K)**2 + BY(I,J,K)**2 + BZ(I,J,K)**2)*IRHO
            CMAX = SQRT(CS2 + CA2)
            DTTEMP = (VX + CMAX)*IDX + (VY + CMAX)*IDY + (VZ + CMAX)*IDZ
            DTTEMPMAX = MAX(DTTEMP, DTTEMPMAX)
          END DO
        END DO
      END DO

C     Calculate the new timestep...
      DT = CFL / DTTEMPMAX

D     PRINT *, '<- SETDT'
      RETURN

      E N D


      SUBROUTINE ADVANCE_ALL

C     Advance everything to the next time level.

      INTEGER NNX, NNY, NNZ
      PARAMETER (NNX = 22, NNY = 22, NNZ = 22)
      REAL*8 RHO, RVX, RVY, S, GAMMA, BX, BY, BZ, RVZ
      COMMON /PHYSA/  RHO(NNX,NNY,NNZ),
     2                RVX(NNX,NNY,NNZ), RVY(NNX,NNY,NNZ),
     3                S(NNX,NNY,NNZ), GAMMA
      COMMON /PHYSB/  BX(NNX,NNY,NNZ), BY(NNX,NNY,NNZ), BZ(NNX,NNY,NNZ)
      COMMON /PHYSC/  RVZ(NNX,NNY,NNZ)
      REAL*8 RHON, RVXN, RVYN, SN, BXN, BYN, BZN, RVZN
      COMMON /PHYSAN/ RHON(NNX,NNY,NNZ),
     2                RVXN(NNX,NNY,NNZ), RVYN(NNX,NNY,NNZ),
     3                SN(NNX,NNY,NNZ)
```

```
      COMMON /PHYSBN/ BXN(NNX,NNY,NNZ), BYN(NNX,NNY,NNZ),
     2                BZN(NNX,NNY,NNZ)
      COMMON /PHYSCN/ RVZN(NNX,NNY,NNZ)
      INTEGER NX, NY, NZ
      REAL*8 X, Y, Z, DX, DY, DZ, VOL
      COMMON /GRID/   X(NNX), Y(NNY), Z(NNZ),
     2                DX, DY, DZ, VOL, NX, NY, NZ
      INTEGER IT, ITMAX
      REAL*8 CFL, DT, TIME, TMAX
      COMMON /TIME/   DT, IT, ITMAX, TIME, TMAX, CFL
      INTEGER I, J, K
      REAL*8 DTD2, FLX(NNX,NNY,NNZ), FLY(NNX,NNY,NNZ), FLZ(NNX,NNY,NNZ),
     2       SOU(NNX,NNY,NNZ)

D     PRINT *, '-> ADVANCE_ALL'

      DTD2 = 0.5D0*DT

C     Predictor step...

      CALL VELOCITY (RHO, RVX, RVY, RVZ, DTD2, 1)

C     ***NOTE: Be careful of the order in which the equations are solved...

      CALL FLUX_N_SOURCE ('RHO', DTD2, FLX, FLY, FLZ, SOU,
     2                    RHO, RVX, RVY, RVZ, S, GAMMA, BX, BY, BZ)
      CALL ADVANCE (RHO, RHON, FLX, FLY, FLZ, SOU, 1, 1, IT)

      CALL FLUX_N_SOURCE ('RVX', DTD2, FLX, FLY, FLZ, SOU,
     2                    RHO, RVX, RVY, RVZ, S, GAMMA, BX, BY, BZ)
      CALL ADVANCE (RVX, RVXN, FLX, FLY, FLZ, SOU, 2, 1, IT)

      CALL FLUX_N_SOURCE ('RVY', DTD2, FLX, FLY, FLZ, SOU,
     2                    RHO, RVX, RVY, RVZ, S, GAMMA, BX, BY, BZ)
      CALL ADVANCE (RVY, RVYN, FLX, FLY, FLZ, SOU, 3, 1, IT)

      CALL FLUX_N_SOURCE ('RVZ', DTD2, FLX, FLY, FLZ, SOU,
     2                    RHO, RVX, RVY, RVZ, S, GAMMA, BX, BY, BZ)
      CALL ADVANCE (RVZ, RVZN, FLX, FLY, FLZ, SOU, 4, 1, IT)

      CALL FLUX_N_SOURCE ('S  ', DTD2, FLX, FLY, FLZ, SOU,
     2                    RHO, RVX, RVY, RVZ, S, GAMMA, BX, BY, BZ)
      CALL ADVANCE (S, SN, FLX, FLY, FLZ, SOU, 5, 1, IT)

      CALL FLUX_N_SOURCE ('BX ', DTD2, FLX, FLY, FLZ, SOU,
     2                    RHO, RVX, RVY, RVZ, S, GAMMA, BX, BY, BZ)
      CALL ADVANCE (BX, BXN, FLX, FLY, FLZ, SOU, 6, 1, IT)

      CALL FLUX_N_SOURCE ('BY ', DTD2, FLX, FLY, FLZ, SOU,
     2                    RHO, RVX, RVY, RVZ, S, GAMMA, BX, BY, BZ)
      CALL ADVANCE (BY, BYN, FLX, FLY, FLZ, SOU, 7, 1, IT)

      CALL FLUX_N_SOURCE ('BZ ', DTD2, FLX, FLY, FLZ, SOU,
     2                    RHO, RVX, RVY, RVZ, S, GAMMA, BX, BY, BZ)
```

```
      CALL ADVANCE (BZ, BZN, FLX, FLY, FLZ, SOU, 8, 1, IT)

C     Corrector step...

      CALL VELOCITY (RHON, RVXN, RVYN, RVZN, DT, 2)

      CALL FLUX_N_SOURCE ('RHO', DT, FLX, FLY, FLZ, SOU,
     2                    RHON, RVXN, RVYN, RVZN, SN, GAMMA,
     3                    BXN, BYN, BZN)
      CALL ADVANCE (RHO, RHON, FLX, FLY, FLZ, SOU, 1, 2, IT)

      CALL FLUX_N_SOURCE ('RVX', DT, FLX, FLY, FLZ, SOU,
     2                    RHON, RVXN, RVYN, RVZN, SN, GAMMA,
     3                    BXN, BYN, BZN)
      CALL ADVANCE (RVX, RVXN, FLX, FLY, FLZ, SOU, 2, 2, IT)

      CALL FLUX_N_SOURCE ('RVY', DT, FLX, FLY, FLZ, SOU,
     2                    RHON, RVXN, RVYN, RVZN, SN, GAMMA,
     3                    BXN, BYN, BZN)
      CALL ADVANCE (RVY, RVYN, FLX, FLY, FLZ, SOU, 3, 2, IT)

      CALL FLUX_N_SOURCE ('RVZ', DT, FLX, FLY, FLZ, SOU,
     2                    RHON, RVXN, RVYN, RVZN, SN, GAMMA,
     3                    BXN, BYN, BZN)
      CALL ADVANCE (RVZ, RVZN, FLX, FLY, FLZ, SOU, 4, 2, IT)

      CALL FLUX_N_SOURCE ('S  ', DT, FLX, FLY, FLZ, SOU,
     2                    RHON, RVXN, RVYN, RVZN, SN, GAMMA,
     3                    BXN, BYN, BZN)
      CALL ADVANCE (S, SN, FLX, FLY, FLZ, SOU, 5, 2, IT)

      CALL FLUX_N_SOURCE ('BX ', DT, FLX, FLY, FLZ, SOU,
     2                    RHON, RVXN, RVYN, RVZN, SN, GAMMA,
     3                    BXN, BYN, BZN)
      CALL ADVANCE (BX, BXN, FLX, FLY, FLZ, SOU, 6, 2, IT)

      CALL FLUX_N_SOURCE ('BY ', DT, FLX, FLY, FLZ, SOU,
     2                    RHON, RVXN, RVYN, RVZN, SN, GAMMA,
     3                    BXN, BYN, BZN)
      CALL ADVANCE (BY, BYN, FLX, FLY, FLZ, SOU, 7, 2, IT)

      CALL FLUX_N_SOURCE ('BZ ', DT, FLX, FLY, FLZ, SOU,
     2                    RHON, RVXN, RVYN, RVZN, SN, GAMMA,
     3                    BXN, BYN, BZN)
      CALL ADVANCE (BZ, BZN, FLX, FLY, FLZ, SOU, 8, 2, IT)

C     Make the new values the current values...
      DO K = 1, NZ
        DO J = 1, NY
          DO I = 1, NX
            RHO(I,J,K) = RHON(I,J,K)
            RVX(I,J,K) = RVXN(I,J,K)
            RVY(I,J,K) = RVYN(I,J,K)
            RVZ(I,J,K) = RVZN(I,J,K)
```

```
            S(I,J,K) = SN(I,J,K)
            BX(I,J,K) = BXN(I,J,K)
            BY(I,J,K) = BYN(I,J,K)
            BZ(I,J,K) = BZN(I,J,K)
          END DO
        END DO
      END DO

D     PRINT *, '<- ADVANCE_ALL'
      RETURN

      E N D


      SUBROUTINE VELOCITY (RHO, RVX, RVY, RVZ, DT, PC)

C     Calculate the velocity-dependent diffusion/antidiffusion coefficients.
C     The values of the coefficients determine the trade-off between
C     numerical diffusion and dispersion in the algorithm (see Appendix B).

      INTEGER NNX, NNY, NNZ
      PARAMETER (NNX = 22, NNY = 22, NNZ = 22)
      REAL*8 ZZA, ZZB, ZZC, ZZD, ZZE
      PARAMETER (ZZA = 1.D0/6.D0, ZZB = 1.D0/3.D0, ZZC = 1.D0/48.D0,
     2           ZZD = 1.D0/2880.D0, ZZE = 1.D0/8.D0)
      INTEGER PC
      REAL*8 RHO(NNX,NNY,NNZ),
     2       RVX(NNX,NNY,NNZ), RVY(NNX,NNY,NNZ), RVZ(NNX,NNY,NNZ), DT
      INTEGER NX, NY, NZ
      REAL*8 X, Y, Z, DX, DY, DZ, VOL
      COMMON /GRID/   X(NNX), Y(NNY), Z(NNZ),
     2                DX, DY, DZ, VOL, NX, NY, NZ
      REAL*8 VXH, VYH, VZH, NUXH, NUYH, NUZH, MUXH, MUYH, MUZH,
     2       GAXH, GAYH, GAZH
      COMMON /VELOC/  VXH(NNX,NNY,NNZ), VYH(NNX,NNY,NNZ),
     2                VZH(NNX,NNY,NNZ), NUXH(NNX,NNY,NNZ),
     3                NUYH(NNX,NNY,NNZ), NUZH(NNX,NNY,NNZ),
     4                MUXH(NNX,NNY,NNZ), MUYH(NNX,NNY,NNZ),
     5                MUZH(NNX,NNY,NNZ), GAXH(NNX,NNY,NNZ),
     6                GAYH(NNX,NNY,NNZ), GAZH(NNX,NNY,NNZ)
      INTEGER NXM1, NYM1, NZM1, OUT_NUM
      REAL*8 IDX, IDY, IDZ, IVOL, I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2
      CHARACTER EQN*3, ID*8
      COMMON /MISC/   OUT_NUM, NXM1, NYM1, NZM1, IDX, IDY, IDZ, IVOL,
     2                I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2,
     3                ID, EQN(8)
      INTEGER I, J, K
      REAL*8 IRHO, CX, CX2, CY, CY2, CZ, CZ2, TEMPX, TEMPY, TEMPZ

D     PRINT *, '-> VELOCITY'

C     Calculate 0.5*velocity...
      DO K = 1, NZ
        DO J = 1, NY
```

```
          DO I = 1, NX
            IRHO = 0.5D0/RHO(I,J,K)
            VXH(I,J,K) = RVX(I,J,K)*IRHO
            VYH(I,J,K) = RVY(I,J,K)*IRHO
            VZH(I,J,K) = RVZ(I,J,K)*IRHO
          END DO
        END DO
      END DO

C     Calculate velocity-dependent diffusion/antidiffusion coefficients...
      IF (PC .EQ. 1) THEN
C       ... for the predictor step...
        DO K = 1, NZM1
          DO J = 1, NYM1
            DO I = 1, NXM1
              CX = ABS(VXH(I,J,K) + VXH(I+1,J,K))*DT*IDX
              CX2 = CX**2
              CY = ABS(VYH(I,J,K) + VYH(I,J+1,K))*DT*IDY
              CY2 = CY**2
              CZ = ABS(VZH(I,J,K) + VZH(I,J,K+1))*DT*IDZ
              CZ2 = CZ**2
              NUXH(I,J,K) = 11.D0/30.D0 - 1.D0/50000.D0
     2                      - 1.D0/300.D0*CX
              NUYH(I,J,K) = 11.D0/30.D0 - 1.D0/50000.D0
     2                      - 1.D0/300.D0*CY
              NUZH(I,J,K) = 11.D0/30.D0 - 1.D0/50000.D0
     2                      - 1.D0/300.D0*CZ
              MUXH(I,J,K) = ZZA - ZZA*CX2
              MUYH(I,J,K) = ZZA - ZZA*CY2
              MUZH(I,J,K) = ZZA - ZZA*CZ2
              GAXH(I,J,K) = ZZA + ZZB*CX2 - NUXH(I,J,K)
              GAYH(I,J,K) = ZZA + ZZB*CY2 - NUYH(I,J,K)
              GAZH(I,J,K) = ZZA + ZZB*CZ2 - NUZH(I,J,K)
            END DO
          END DO
        END DO
      ELSE IF (PC .EQ. 2) THEN
C       ... for the corrector step...
        DO K = 1, NZM1
          DO J = 1, NYM1
            DO I = 1, NXM1
              CX = ABS(VXH(I,J,K) + VXH(I+1,J,K))*DT*IDX
              CX2 = CX**2
              CY = ABS(VYH(I,J,K) + VYH(I,J+1,K))*DT*IDY
              CY2 = CY**2
              CZ = ABS(VZH(I,J,K) + VZH(I,J,K+1))*DT*IDZ
              CZ2 = CZ**2
              TEMPX = 11.D0/30.D0 - 1.D0/50000.D0
     2                - 1.D0/300.D0*(0.5D0*CX)
              TEMPX = SQRT(506880.D0 - (115200.D0 - 29520.D0*CX2)*CX2
     2                     - (1382400.D0 - 345600.D0*CX2)*TEMPX)
              TEMPY = 11.D0/30.D0 - 1.D0/50000.D0
     2                - 1.D0/300.D0*(0.5D0*CY)
              TEMPY = SQRT(506880.D0 - (115200.D0 - 29520.D0*CY2)*CY2
```

```
     2                    - (1382400.D0 - 345600.D0*CY2)*TEMPY)
              TEMPZ = 11.D0/30.D0 - 1.D0/50000.D0
     2              - 1.D0/300.D0*(0.5D0*CZ)
              TEMPZ = SQRT(506880.D0 - (115200.D0 - 29520.D0*CZ2)*CZ2
     2                    - (1382400.D0 - 345600.D0*CZ2)*TEMPZ)
              NUXH(I,J,K) = ZZA + ZZC*CX2 - ZZD*TEMPX
              NUYH(I,J,K) = ZZA + ZZC*CY2 - ZZD*TEMPY
              NUZH(I,J,K) = ZZA + ZZC*CZ2 - ZZD*TEMPZ
              MUXH(I,J,K) = ZZA - 5.D0*ZZC*CX2 + ZZD*TEMPX
              MUYH(I,J,K) = ZZA - 5.D0*ZZC*CY2 + ZZD*TEMPY
              MUZH(I,J,K) = ZZA - 5.D0*ZZC*CZ2 + ZZD*TEMPZ
              GAXH(I,J,K) = -ZZE*CX2 + 2.D0*ZZD*TEMPX
              GAYH(I,J,K) = -ZZE*CY2 + 2.D0*ZZD*TEMPY
              GAZH(I,J,K) = -ZZE*CZ2 + 2.D0*ZZD*TEMPZ
            END DO
          END DO
        END DO
      END IF

D     PRINT *, '<- VELOCITY'
      RETURN

      E N D


      SUBROUTINE FLUX_N_SOURCE (EQN_NAME, DT, FLX, FLY, FLZ, SOU,
     2                          RHO, RVX, RVY, RVZ, S, GAMMA,
     3                          BX, BY, BZ)

C     Calculate the fluxes at the centres of the cells.

      INTEGER NNX, NNY, NNZ
      PARAMETER (NNX = 22, NNY = 22, NNZ = 22)
      CHARACTER EQN_NAME*3
      REAL*8 DT, FLX(NNX,NNY,NNZ), FLY(NNX,NNY,NNZ), FLZ(NNX,NNY,NNZ),
     2       SOU(NNX,NNY,NNZ), RHO(NNX,NNY,NNZ), RVX(NNX,NNY,NNZ),
     3       RVY(NNX,NNY,NNZ), RVZ(NNX,NNY,NNZ), S(NNX,NNY,NNZ), GAMMA,
     4       BX(NNX,NNY,NNZ), BY(NNX,NNY,NNZ), BZ(NNX,NNY,NNZ)
      INTEGER NX, NY, NZ
      REAL*8 X, Y, Z, DX, DY, DZ, VOL
      COMMON /GRID/   X(NNX), Y(NNY), Z(NNZ),
     2                DX, DY, DZ, VOL, NX, NY, NZ
      REAL*8 VXH, VYH, VZH, NUXH, NUYH, NUZH, MUXH, MUYH, MUZH,
     2       GAXH, GAYH, GAZH
      COMMON /VELOC/  VXH(NNX,NNY,NNZ), VYH(NNX,NNY,NNZ),
     2                VZH(NNX,NNY,NNZ), NUXH(NNX,NNY,NNZ),
     3                NUYH(NNX,NNY,NNZ), NUZH(NNX,NNY,NNZ),
     4                MUXH(NNX,NNY,NNZ), MUYH(NNX,NNY,NNZ),
     5                MUZH(NNX,NNY,NNZ), GAXH(NNX,NNY,NNZ),
     6                GAYH(NNX,NNY,NNZ), GAZH(NNX,NNY,NNZ)
      INTEGER NXM1, NYM1, NZM1, OUT_NUM
      REAL*8 IDX, IDY, IDZ, IVOL, I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2
      CHARACTER EQN*3, ID*8
      COMMON /MISC/   OUT_NUM, NXM1, NYM1, NZM1, IDX, IDY, IDZ, IVOL,
```

```
     2               I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2,
     3               ID, EQN(8)
      INTEGER I, J, K
      REAL*8 DXDY, DXDZ, DYDZ, GAMM1, PTOT(NNX,NNY,NNZ)

D     PRINT *, '-> FLUX_N_SOURCE'

      GAMM1 = GAMMA - 1.D0
      DXDY = DX*DY
      DXDZ = DX*DZ
      DYDZ = DY*DZ

      DO I = 1, 8
        IF (EQN_NAME .EQ. EQN(I))  GO TO 1
      END DO
      PRINT *, ' Equation name "', EQN_NAME, '" not recognized...'
      STOP

 1    GO TO (110, 120, 130, 140, 150, 160, 170, 180) I

C     Note that the nozero source terms calculated below, SOU(I,J,K), are
C     bad for I, J and K = 1, but those array elements are never used, so
C     everything's alright, though your compiler may not think so.

C     Continuity equation...
 110  CALL CONV_FLUX (RHO, FLX, FLY, FLZ, DT)
      DO K = 1, NZM1
        DO J = 1, NYM1
          DO I = 1, NXM1
            SOU(I,J,K) = 0.D0
          END DO
        END DO
      END DO

D     PRINT *, '<- FLUX_N_SOURCE'
      RETURN

C     Calculate the total pressure for use here and in the Y and Z-momentum equation...
 120  DO K = 1, NZ
        DO J = 1, NY
          DO I = 1, NX
            PTOT(I,J,K) = S(I,J,K) + 0.5D0*(
     2                    BX(I,J,K)**2 + BY(I,J,K)**2 + BZ(I,J,K)**2
     3                    )
          END DO
        END DO
      END DO

C     X-momentum equation...
      CALL CONV_FLUX (RVX, FLX, FLY, FLZ, DT)
      DO K = 1, NZM1
        DO J = 1, NYM1
          DO I = 1, NXM1
            SOU(I,J,K) = 0.5D0*(
```

```
     2                  (PTOT(I-1,J,K) - PTOT(I+1,J,K))*DYDZ +
     3                  BX(I,J,K)*(BX(I+1,J,K) - BX(I-1,J,K))*DYDZ +
     4                  BY(I,J,K)*(BX(I,J+1,K) - BX(I,J-1,K))*DXDZ +
     5                  BZ(I,J,K)*(BX(I,J,K+1) - BX(I,J,K-1))*DXDY
     6                  )*DT
         END DO
       END DO
      END DO

D     PRINT *, '<- FLUX_N_SOURCE'
      RETURN

C     Y-momentum equation...
 130  CALL CONV_FLUX (RVY, FLX, FLY, FLZ, DT)
      DO K = 1, NZM1
        DO J = 1, NYM1
          DO I = 1, NXM1
            SOU(I,J,K) = 0.5D0*(
     2                  (PTOT(I,J-1,K) - PTOT(I,J+1,K))*DXDZ +
     3                  BX(I,J,K)*(BY(I+1,J,K) - BY(I-1,J,K))*DYDZ +
     4                  BY(I,J,K)*(BY(I,J+1,K) - BY(I,J-1,K))*DXDZ +
     5                  BZ(I,J,K)*(BY(I,J,K+1) - BY(I,J,K-1))*DXDY
     6                  )*DT
         END DO
       END DO
      END DO

D     PRINT *, '<- FLUX_N_SOURCE'
      RETURN

C     Z-momentum equation...
 140  CALL CONV_FLUX (RVZ, FLX, FLY, FLZ, DT)
      DO K = 1, NZM1
        DO J = 1, NYM1
          DO I = 1, NXM1
            SOU(I,J,K) = 0.5D0*(
     2                  (PTOT(I,J,K-1) - PTOT(I,J,K+1))*DXDY +
     3                  BX(I,J,K)*(BZ(I+1,J,K) - BZ(I-1,J,K))*DYDZ +
     4                  BY(I,J,K)*(BZ(I,J+1,K) - BZ(I,J-1,K))*DXDZ +
     5                  BZ(I,J,K)*(BZ(I,J,K+1) - BZ(I,J,K-1))*DXDY
     6                  )*DT
         END DO
       END DO
      END DO

D     PRINT *, '<- FLUX_N_SOURCE'
      RETURN

C     Thermodynamics equation...
 150  CALL CONV_FLUX (S, FLX, FLY, FLZ, DT)
      DO K = 1, NZM1
        DO J = 1, NYM1
          DO I = 1, NXM1
            SOU(I,J,K) = -GAMM1*S(I,J,K)*(
```

```
     2                  (VXH(I+1,J,K) - VXH(I-1,J,K))*DYDZ +
     3                  (VYH(I,J+1,K) - VYH(I,J-1,K))*DXDZ +
     4                  (VZH(I,J,K+1) - VZH(I,J,K-1))*DXDY
     5                  )*DT
          END DO
        END DO
       END DO

D      PRINT *, '<- FLUX_N_SOURCE'
       RETURN

C      X-magnetic field equation...
 160   CALL CONV_FLUX (BX, FLX, FLY, FLZ, DT)
       DO K = 1, NZM1
         DO J = 1, NYM1
           DO I = 1, NXM1
             SOU(I,J,K) = (BX(I,J,K)*(VXH(I+1,J,K) - VXH(I-1,J,K))*DYDZ +
     2                     BY(I,J,K)*(VXH(I,J+1,K) - VXH(I,J-1,K))*DXDZ +
     3                     BZ(I,J,K)*(VXH(I,J,K+1) - VXH(I,J,K-1))*DXDY
     4                    )*DT
          END DO
        END DO
       END DO

D      PRINT *, '<- FLUX_N_SOURCE'
       RETURN

C      Y-magnetic field equation...
 170   CALL CONV_FLUX (BY, FLX, FLY, FLZ, DT)
       DO K = 1, NZM1
         DO J = 1, NYM1
           DO I = 1, NXM1
             SOU(I,J,K) = (BX(I,J,K)*(VYH(I+1,J,K) - VYH(I-1,J,K))*DYDZ +
     2                     BY(I,J,K)*(VYH(I,J+1,K) - VYH(I,J-1,K))*DXDZ +
     3                     BZ(I,J,K)*(VYH(I,J,K+1) - VYH(I,J,K-1))*DXDY
     4                    )*DT
          END DO
        END DO
       END DO

D      PRINT *, '<- FLUX_N_SOURCE'
       RETURN

C      Z-magnetic field equation...
 180   CALL CONV_FLUX (BZ, FLX, FLY, FLZ, DT)
       DO K = 1, NZM1
         DO J = 1, NYM1
           DO I = 1, NXM1
             SOU(I,J,K) = (BX(I,J,K)*(VZH(I+1,J,K) - VZH(I-1,J,K))*DYDZ +
     2                     BY(I,J,K)*(VZH(I,J+1,K) - VZH(I,J-1,K))*DXDZ +
     3                     BZ(I,J,K)*(VZH(I,J,K+1) - VZH(I,J,K-1))*DXDY
     4                    )*DT
          END DO
        END DO
```

```
      END DO

D     PRINT *, '<- FLUX_N_SOURCE'
      RETURN

      E N D


      SUBROUTINE CONV_FLUX (F, FLX, FLY, FLZ, DT)

C     Calculate the convective fluxes.

      INTEGER NNX, NNY, NNZ
      PARAMETER (NNX = 22, NNY = 22, NNZ = 22)
      REAL*8 DT, F(NNX,NNY,NNZ),
     2       FLX(NNX,NNY,NNZ), FLY(NNX,NNY,NNZ), FLZ(NNX,NNY,NNZ)
      INTEGER NX, NY, NZ
      REAL*8 X, Y, Z, DX, DY, DZ, VOL
      COMMON /GRID/   X(NNX), Y(NNY), Z(NNZ),
     2                DX, DY, DZ, VOL, NX, NY, NZ
      REAL*8 VXH, VYH, VZH, NUXH, NUYH, NUZH, MUXH, MUYH, MUZH,
     2       GAXH, GAYH, GAZH
      COMMON /VELOC/  VXH(NNX,NNY,NNZ), VYH(NNX,NNY,NNZ),
     2                VZH(NNX,NNY,NNZ), NUXH(NNX,NNY,NNZ),
     3                NUYH(NNX,NNY,NNZ), NUZH(NNX,NNY,NNZ),
     4                MUXH(NNX,NNY,NNZ), MUYH(NNX,NNY,NNZ),
     5                MUZH(NNX,NNY,NNZ), GAXH(NNX,NNY,NNZ),
     6                GAYH(NNX,NNY,NNZ), GAZH(NNX,NNY,NNZ)
      INTEGER NXM1, NYM1, NZM1, OUT_NUM
      REAL*8 IDX, IDY, IDZ, IVOL, I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2
      CHARACTER EQN*3, ID*8
      COMMON /MISC/   OUT_NUM, NXM1, NYM1, NZM1, IDX, IDY, IDZ, IVOL,
     2                I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2,
     3                ID, EQN(8)
      INTEGER I, J, K
      REAL*8 DXDYDT, DXDZDT, DYDZDT

D     PRINT *, '-> CONV_FLUX'

C     Use ZIP fluxes [Zalesak, J. Comp. Phys., vol. 40, pp. 497, 1981]...
      DXDYDT = DX*DY*DT
      DXDZDT = DX*DZ*DT
      DYDZDT = DY*DZ*DT
      DO K = 1, NZM1
        DO J = 1, NYM1
          DO I = 1, NXM1
            FLX(I,J,K) = DYDZDT*(
     2                   F(I,J,K)*VXH(I+1,J,K) +
     3                   F(I+1,J,K)*VXH(I,J,K)
     4                   )
            FLY(I,J,K) = DXDZDT*(
     2                   F(I,J,K)*VYH(I,J+1,K) +
     3                   F(I,J+1,K)*VYH(I,J,K)
     4                   )
```

```
            FLZ(I,J,K) = DXDYDT*(
     2                   F(I,J,K)*VZH(I,J,K+1) +
     3                   F(I,J,K+1)*VZH(I,J,K)
     4                   )
          END DO
        END DO
      END DO

D     PRINT *, '<- CONV_FLUX'
      RETURN

      E N D


      SUBROUTINE ADVANCE (F, FN, FLX, FLY, FLZ, SOU, IVAR, PC, IT)

C     Advance F and then diffuse, limit and antidiffuse it.

      INTEGER IT, IVAR, NNX, NNY, NNZ
      PARAMETER (NNX = 22, NNY = 22, NNZ = 22)
      REAL*8 F(NNX,NNY,NNZ), FN(NNX,NNY,NNZ), FLX(NNX,NNY,NNZ),
     2       FLY(NNX,NNY,NNZ), FLZ(NNX,NNY,NNZ), SOU(NNX,NNY,NNZ)
      REAL*8 VXH, VYH, VZH, NUXH, NUYH, NUZH, MUXH, MUYH, MUZH,
     2       GAXH, GAYH, GAZH
      COMMON /VELOC/  VXH(NNX,NNY,NNZ), VYH(NNX,NNY,NNZ),
     2                VZH(NNX,NNY,NNZ), NUXH(NNX,NNY,NNZ),
     3                NUYH(NNX,NNY,NNZ), NUZH(NNX,NNY,NNZ),
     4                MUXH(NNX,NNY,NNZ), MUYH(NNX,NNY,NNZ),
     5                MUZH(NNX,NNY,NNZ), GAXH(NNX,NNY,NNZ),
     6                GAYH(NNX,NNY,NNZ), GAZH(NNX,NNY,NNZ)
      INTEGER NX, NY, NZ
      REAL*8 X, Y, Z, DX, DY, DZ, VOL
      COMMON /GRID/   X(NNX), Y(NNY), Z(NNZ),
     2                DX, DY, DZ, VOL, NX, NY, NZ
      INTEGER NXM1, NYM1, NZM1, OUT_NUM
      REAL*8 IDX, IDY, IDZ, IVOL, I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2
      CHARACTER EQN*3, ID*8
      COMMON /MISC/   OUT_NUM, NXM1, NYM1, NZM1, IDX, IDY, IDZ, IVOL,
     2                I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2,
     3                ID, EQN(8)
      INTEGER I, J, K, PC
      REAL*8 FH(NNX,NNY,NNZ), FTD(NNX,NNY,NNZ)

D     PRINT *, '-> ADVANCE'

C     Copy the half-step F for use in antidiffusion
C     when doing the corrector step...
      IF (PC .EQ. 2) THEN
        DO K = 1, NZ
          DO J = 1, NY
            DO I = 1, NX
              FH(I,J,K) = FN(I,J,K)
            END DO
          END DO
```

```
        END DO
      END IF

C     Advance F...
      DO K = 2, NZM1
        DO J = 2, NYM1
          DO I = 2, NXM1
            FN(I,J,K) = F(I,J,K) +
     2                 ((FLX(I-1,J,K) - FLX(I,J,K)) +
     3                  (FLY(I,J-1,K) - FLY(I,J,K)) +
     4                  (FLZ(I,J,K-1) - FLZ(I,J,K))
     5                  + SOU(I,J,K))*IVOL
          END DO
        END DO
      END DO

C     Prediffuse:
C      the initial F when doing the predictor step...
      IF (PC .EQ. 1) THEN
        DO K = 1, NZM1
          DO J = 1, NYM1
            DO I = 1, NXM1
              FLX(I,J,K) = GAXH(I,J,K)*(F(I+1,J,K) - F(I,J,K))
              FLY(I,J,K) = GAYH(I,J,K)*(F(I,J+1,K) - F(I,J,K))
              FLZ(I,J,K) = GAZH(I,J,K)*(F(I,J,K+1) - F(I,J,K))
            END DO
          END DO
        END DO
C      the half-step F when doing the corrector step...
      ELSE IF (PC .EQ. 2) THEN
        DO K = 1, NZM1
          DO J = 1, NYM1
            DO I = 1, NXM1
              FLX(I,J,K) = GAXH(I,J,K)*(FH(I+1,J,K) - FH(I,J,K))
              FLY(I,J,K) = GAYH(I,J,K)*(FH(I,J+1,K) - FH(I,J,K))
              FLZ(I,J,K) = GAZH(I,J,K)*(FH(I,J,K+1) - FH(I,J,K))
            END DO
          END DO
        END DO
      END IF
      DO K = 2, NZM1
        DO J = 2, NYM1
          DO I = 2, NXM1
            FN(I,J,K) = FN(I,J,K) + (FLX(I,J,K) - FLX(I-1,J,K)) +
     2                              (FLY(I,J,K) - FLY(I,J-1,K)) +
     3                              (FLZ(I,J,K) - FLZ(I,J,K-1))
          END DO
        END DO
      END DO

      IF (PC .EQ. 1) THEN
        CALL BCOND (FN, F, IVAR)
      ELSE
        CALL BCOND (FN, FH, IVAR)
```

```
      END IF

C     Calculate the diffusive fluxes...
      DO K = 1, NZM1
        DO J = 1, NYM1
          DO I = 1, NXM1
            FLX(I,J,K) = NUXH(I,J,K)*(F(I+1,J,K) - F(I,J,K))
            FLY(I,J,K) = NUYH(I,J,K)*(F(I,J+1,K) - F(I,J,K))
            FLZ(I,J,K) = NUZH(I,J,K)*(F(I,J,K+1) - F(I,J,K))
          END DO
        END DO
      END DO

C     Diffuse the advanced F...
      DO K = 2, NZM1
        DO J = 2, NYM1
          DO I = 2, NXM1
            FTD(I,J,K) = FN(I,J,K) + (FLX(I,J,K) - FLX(I-1,J,K)) +
     2                               (FLY(I,J,K) - FLY(I,J-1,K)) +
     3                               (FLZ(I,J,K) - FLZ(I,J,K-1))
          END DO
        END DO
      END DO

      IF (PC .EQ. 1) THEN
        CALL BCOND (FTD, F, IVAR)
      ELSE
        CALL BCOND (FTD, FH, IVAR)
      END IF

C     Calculate the antidiffusive fluxes...
      DO K = 1, NZ
        DO J = 1, NY
          DO I = 1, NXM1
            FLX(I,J,K) = MUXH(I,J,K)*VOL*(FN(I+1,J,K) - FN(I,J,K))
          END DO
        END DO
      END DO
      DO K = 1, NZ
        DO J = 1, NYM1
          DO I = 1, NX
            FLY(I,J,K) = MUYH(I,J,K)*VOL*(FN(I,J+1,K) - FN(I,J,K))
          END DO
        END DO
      END DO
      DO K = 1, NZM1
        DO J = 1, NY
          DO I = 1, NX
            FLZ(I,J,K) = MUZH(I,J,K)*VOL*(FN(I,J,K+1) - FN(I,J,K))
          END DO
        END DO
      END DO

C     Prelimit the antidiffusive fluxes (in all directions simultaneously)...
```

```
      CALL PRELIMIT (FLX, FLY, FLZ, FTD, NX, NY, NZ)
C     Limit the antidiffusive fluxes (in all directions simultaneously) and
C     do it so that the errors of the HI and LO limiters cancel (somewhat)...
      IF (MOD(PC + IT, 2) .EQ. 0) THEN
        CALL HI_LIMIT (FLX, FLY, FLZ, FTD, VOL, F, IVAR)
        CALL LO_LIMIT (FLX, FLY, FLZ, FTD, VOL, F, IVAR)
      ELSE
        CALL LO_LIMIT (FLX, FLY, FLZ, FTD, VOL, F, IVAR)
        CALL HI_LIMIT (FLX, FLY, FLZ, FTD, VOL, F, IVAR)
      END IF

C     Use the limited fluxes to calculate the best FN...
      DO K = 2, NZM1
        DO J = 2, NYM1
          DO I = 2, NXM1
            FN(I,J,K) = FTD(I,J,K) +
     2                 ((FLX(I-1,J,K) - FLX(I,J,K)) +
     3                  (FLY(I,J-1,K) - FLY(I,J,K)) +
     4                  (FLZ(I,J,K-1) - FLZ(I,J,K))
     5                  )*IVOL
          END DO
        END DO
      END DO

      IF (PC .EQ. 1) THEN
        CALL BCOND (FN, F, IVAR)
      ELSE
        CALL BCOND (FN, FH, IVAR)
      END IF

D     PRINT *, '<- ADVANCE'
      RETURN

      E N D


      SUBROUTINE BCOND (F, FOLD, IVAR)

C     Set the boundary conditions.

      INTEGER IVAR, NNX, NNY, NNZ
      PARAMETER (NNX = 22, NNY = 22, NNZ = 22)
      REAL*8 F(NNX,NNY,NNZ), FOLD(NNX,NNY,NNZ)
      REAL*8 BC
      COMMON /BC/     BC(6,8)
      INTEGER NX, NY, NZ
      REAL*8 X, Y, Z, DX, DY, DZ, VOL
      COMMON /GRID/   X(NNX), Y(NNY), Z(NNZ),
     2                DX, DY, DZ, VOL, NX, NY, NZ
      INTEGER IT, ITMAX
      REAL*8 CFL, DT, TIME, TMAX
      COMMON /TIME/   DT, IT, ITMAX, TIME, TMAX, CFL
      INTEGER NXM1, NYM1, NZM1, OUT_NUM
      REAL*8 IDX, IDY, IDZ, IVOL, I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2
```

```
      CHARACTER EQN*3, ID*8
      COMMON /MISC/   OUT_NUM, NXM1, NYM1, NZM1, IDX, IDY, IDZ, IVOL,
     2                I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2,
     3                ID, EQN(8)
      INTEGER I, J, K
      REAL*8 PBCX, PBCY, PBCZ, RBCX, RBCY, RBCZ

D     PRINT *, '-> BCOND'

      RBCX = BC(1,IVAR)
      RBCY = BC(3,IVAR)
      RBCZ = BC(5,IVAR)
      PBCX = 1.D0 - RBCX**2
      PBCY = 1.D0 - RBCY**2
      PBCZ = 1.D0 - RBCZ**2

C     x boundary conditions...
      DO K = 2, NZM1
        DO J = 2, NYM1
          F(1,J,K) = PBCX*F(NXM1,J,K) + RBCX*F(2,J,K)
          F(NX,J,K) = RBCX*F(NXM1,J,K) + PBCX*F(2,J,K)
        END DO
      END DO

C     y boundary conditions...
      DO K = 1, NZ
        DO I = 1, NX
          F(I,1,K) = PBCY*F(I,NYM1,K) + RBCY*F(I,2,K)
          F(I,NY,K) = RBCY*F(I,NYM1,K) + PBCY*F(I,2,K)
        END DO
      END DO

C     z boundary conditions...
      DO J = 1, NY
        DO I = 1, NX
          F(I,J,1) = PBCZ*F(I,J,NZM1) + RBCZ*F(I,J,2)
          F(I,J,NZ) = RBCZ*F(I,J,NZM1) + PBCZ*F(I,J,2)
        END DO
      END DO

D     PRINT *, '<- BCOND'
      RETURN

      END


      SUBROUTINE DATA_OUT

C     Write a snapshot of the simulation to a file.

      INTEGER NNX, NNY, NNZ
      PARAMETER (NNX = 22, NNY = 22, NNZ = 22)
      REAL*8 RHO, RVX, RVY, S, GAMMA, BX, BY, BZ, RVZ
      COMMON /PHYSA/  RHO(NNX,NNY,NNZ),
```

```
     2               RVX(NNX,NNY,NNZ), RVY(NNX,NNY,NNZ),
     3               S(NNX,NNY,NNZ), GAMMA
      COMMON /PHYSB/  BX(NNX,NNY,NNZ), BY(NNX,NNY,NNZ), BZ(NNX,NNY,NNZ)
      COMMON /PHYSC/  RVZ(NNX,NNY,NNZ)
      INTEGER NX, NY, NZ
      REAL*8 X, Y, Z, DX, DY, DZ, VOL
      COMMON /GRID/   X(NNX), Y(NNY), Z(NNZ),
     2               DX, DY, DZ, VOL, NX, NY, NZ
      INTEGER IT, ITMAX
      REAL*8 CFL, DT, TIME, TMAX
      COMMON /TIME/   DT, IT, ITMAX, TIME, TMAX, CFL
      INTEGER NXM1, NYM1, NZM1, OUT_NUM
      REAL*8 IDX, IDY, IDZ, IVOL, I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2
      CHARACTER EQN*3, ID*8
      COMMON /MISC/   OUT_NUM, NXM1, NYM1, NZM1, IDX, IDY, IDZ, IVOL,
     2               I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2,
     3               ID, EQN(8)
      INTEGER I, J, K, UNIT_LEN
      CHARACTER CHAR_OUT_NUM*2, REMOVE_SPACES*80, UNIT*80

D     PRINT *, '-> DATA_OUT'

C     Name of file to write to... make sure that there are no spaces...
      WRITE (CHAR_OUT_NUM, '(I2.2)') OUT_NUM
      UNIT = ID // '-' // CHAR_OUT_NUM // '.dat'
      UNIT = REMOVE_SPACES(UNIT)
      UNIT_LEN = INDEX(UNIT, ' ') - 1
      IF (UNIT_LEN .EQ. -1)  UNIT_LEN = LEN(UNIT)
      OPEN (9, FILE = UNIT(1:UNIT_LEN), STATUS = 'NEW',
     2                                  FORM = 'UNFORMATTED')
      PRINT *, ' Snapshot file ' // UNIT(1:UNIT_LEN) // ' open...'

      WRITE (9) NX, NY, NZ, NNX, NNY, NNZ
      WRITE (9) X, Y, Z
      WRITE (9) TIME, IT, GAMMA
      CALL WRITE_R4 (RHO, NX, NY, NZ)
      CALL WRITE_R4 (RVX, NX, NY, NZ)
      CALL WRITE_R4 (RVY, NX, NY, NZ)
      CALL WRITE_R4 (RVZ, NX, NY, NZ)
      PRINT *, '  ...writing...'
      CALL WRITE_R4 (S,   NX, NY, NZ)
      CALL WRITE_R4 (BX,  NX, NY, NZ)
      CALL WRITE_R4 (BY,  NX, NY, NZ)
      CALL WRITE_R4 (BZ,  NX, NY, NZ)
      CLOSE (9)
      PRINT *, '    ...done.'
      PRINT *, ' TIME =', REAL(TIME), '  DT =', REAL(DT), '  IT =', IT
      PRINT *, ' '

      OUT_NUM = OUT_NUM + 1

D     PRINT *, '<- DATA_OUT'
      RETURN
```

```
      E N D


      SUBROUTINE WRITE_R4 (F, NX, NY, NZ)

C     Write out F in space-saving REAL*4 unformatted form.

      INTEGER NNX, NNY, NNZ
      PARAMETER (NNX = 22, NNY = 22, NNZ = 22)
      INTEGER I, J, K, NX, NY, NZ
      REAL*8 F(NNX,NNY,NNZ)
      REAL*4 F4(NNX,NNY,NNZ)

D     PRINT *, '-> WRITE_R4'

      DO K = 1, NNZ
        DO J = 1, NNY
          DO I = 1, NNX
            F4(I,J,K) = 0.
          END DO
        END DO
      END DO
      DO K = 1, NZ
        DO J = 1, NY
          DO I = 1, NX
            F4(I,J,K) = REAL(F(I,J,K))
          END DO
        END DO
      END DO

      WRITE (9) F4

D     PRINT *, '<- WRITE_R4'
      RETURN

      E N D


      SUBROUTINE PRINT (F, NAME, STEP, ILO, IHI, JLO, JHI, KLO, KHI)

C     Print the array F. Be sure that IHI - ILO and JHI - JLO are .LE. 98.
C     CALL PRINT (RHO, ' RHO', IT, 2, 6, 2, 6, 14, 14)


      INTEGER NNX, NNY, NNZ
      PARAMETER (NNX = 22, NNY = 22, NNZ = 22)
      INTEGER I, ILO, IHI, J, JLO, JHI, K, KLO, KHI, STEP
      REAL*8 F(NNX,NNY,NNZ)
      CHARACTER COL_NUM*2, FMTA*26, FMTB*21, NAME*4, CSTEP*6

D     PRINT *, '-> PRINT'

      WRITE (CSTEP, '(I6)') STEP
      WRITE (12, *) NAME // ' at step number ' // CSTEP
```

```
      WRITE (COL_NUM, '(I2)') IHI - ILO + 1
      FMTA = '("K=", I2, ' // COL_NUM // '(7X, I2, 7X))'
      FMTB = '(1P, I2, 2X, ' // COL_NUM // 'G16.8)'

      DO K = KHI, KLO, -1
        WRITE (12, FMTA) K, (I, I=ILO,IHI)
        WRITE (12, FMTB) (J, (F(I,J,K), I=ILO,IHI), J=JHI,JLO,-1)
      END DO

      WRITE (12, *) ' '

D     PRINT *, '<- PRINT'
      RETURN

      E N D


      SUBROUTINE FINISH

C     Print representative subsections of the arrays
C     and then stop the simulation.

      INTEGER NNX, NNY, NNZ
      PARAMETER (NNX = 22, NNY = 22, NNZ = 22)
      REAL*8 RHO, RVX, RVY, S, GAMMA, BX, BY, BZ, RVZ
      COMMON /PHYSA/  RHO(NNX,NNY,NNZ),
     2                RVX(NNX,NNY,NNZ), RVY(NNX,NNY,NNZ),
     3                S(NNX,NNY,NNZ), GAMMA
      COMMON /PHYSB/  BX(NNX,NNY,NNZ), BY(NNX,NNY,NNZ), BZ(NNX,NNY,NNZ)
      COMMON /PHYSC/  RVZ(NNX,NNY,NNZ)
      INTEGER IT, ITMAX
      REAL*8 CFL, DT, TIME, TMAX
      COMMON /TIME/   DT, IT, ITMAX, TIME, TMAX, CFL

      CALL PRINT (RHO, ' RHO', IT, 2, 6, 2, 6, 14, 14)
      CALL PRINT (RVX, ' RVX', IT, 2, 6, 2, 6, 14, 14)
      CALL PRINT (RVY, ' RVY', IT, 2, 6, 2, 6, 14, 14)
      CALL PRINT (RVZ, ' RVZ', IT, 2, 6, 2, 6, 14, 14)
      CALL PRINT (S,   '   S', IT, 2, 6, 2, 6, 14, 14)
      CALL PRINT (BX,  '  BX', IT, 2, 6, 2, 6, 14, 14)
      CALL PRINT (BY,  '  BY', IT, 2, 6, 2, 6, 14, 14)
      CALL PRINT (BZ,  '  BZ', IT, 2, 6, 2, 6, 14, 14)

      STOP '... simulation done.'

      E N D


      CHARACTER*(*) FUNCTION REMOVE_SPACES (STRING)

      CHARACTER STRING*(*)
      INTEGER LENGTH, SPACE
```

```
      LENGTH = LEN(STRING)

      SPACE = INDEX(STRING, ' ')
      DO WHILE (SPACE .NE. 0 .AND. SPACE .LE. LENGTH)
        IF (SPACE .EQ. 1) THEN
          STRING = STRING(SPACE+1:LENGTH)
        ELSE IF (SPACE .LT. LENGTH) THEN
          STRING = STRING(1:SPACE-1) // STRING(SPACE+1:LENGTH)
        END IF
        LENGTH = LENGTH - 1
        SPACE = INDEX(STRING, ' ')
      END DO

      REMOVE_SPACES = STRING(1:LENGTH)

      RETURN

      E N D


      BLOCK DATA COMMON_INIT

      INTEGER NNX, NNY, NNZ
      PARAMETER (NNX = 22, NNY = 22, NNZ = 22)
      REAL*8 BC
      COMMON /BC/     BC(6,8)
      INTEGER NX, NY, NZ
      REAL*8 X, Y, Z, DX, DY, DZ, VOL
      COMMON /GRID/   X(NNX), Y(NNY), Z(NNZ),
     2                DX, DY, DZ, VOL, NX, NY, NZ
      INTEGER IT, ITMAX
      REAL*8 CFL, DT, TIME, TMAX
      COMMON /TIME/   DT, IT, ITMAX, TIME, TMAX, CFL
      LOGICAL PRLIM, PROLD
      COMMON /LIMIT/  PRLIM, PROLD
      INTEGER NXM1, NYM1, NZM1, OUT_NUM
      REAL*8 IDX, IDY, IDZ, IVOL, I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2
      CHARACTER EQN*3, ID*8
      COMMON /MISC/   OUT_NUM, NXM1, NYM1, NZM1, IDX, IDY, IDZ, IVOL,
     2                I2DX, I2DY, I2DZ, IDX2, IDY2, IDZ2,
     3                ID, EQN(8)

C     Model ID:
      DATA ID      / 'thesis' /
C     Number of the snapshot file to start from (make nonzero only for model restarts):
      DATA OUT_NUM / 0 /
C     Number of cells in grid:
      DATA NX      / 22 /,
     2     NY      / 22 /,
     3     NZ      / 22 /
C     Define the cell size:
      DATA DX      / 0.05D0 /,
     2     DY      / 0.05D0 /,
     3     DZ      / 0.05D0 /
```

```
C     Starting time, maximum time, initial timestep size:
      DATA TIME    / 0.D0 /,
     2     TMAX    / 0.4D0 /,
     3     DT      / 0.25D-3 /
C     Minimum, desired, and maximum Courant numbers and max. # of iterations:
      DATA CFL     / 0.25D0 /,
     2     ITMAX   / 500 /
C     Switches controlling how the limiter is run:
      DATA PRLIM   / .TRUE. /,
     2     PROLD   / .TRUE. /
C     Equation/variable names:
      DATA EQN     / 'RHO', 'RVX', 'RVY', 'RVZ', 'S  ',
     2               'BX ', 'BY ', 'BZ ' /
C     Define b.c.:     xlo,   xhi,   ylo,   yhi,   zlo,   zhi (xyzlo = xyzhi)
      DATA BC      /  1.D0,  1.D0,  1.D0,  1.D0,  0.D0,  0.D0,          rho
     2               -1.D0, -1.D0,  1.D0,  1.D0,  0.D0,  0.D0,          rvx
     3                1.D0,  1.D0, -1.D0, -1.D0,  0.D0,  0.D0,          rvy
     4                1.D0,  1.D0,  1.D0,  1.D0,  0.D0,  0.D0,          rvz
     5                1.D0,  1.D0,  1.D0,  1.D0,  0.D0,  0.D0,          s
     6               -1.D0, -1.D0,  1.D0,  1.D0,  0.D0,  0.D0,          bx
     7                1.D0,  1.D0, -1.D0, -1.D0,  0.D0,  0.D0,          by
     8                1.D0,  1.D0,  1.D0,  1.D0,  0.D0,  0.D0 /         bz

      E N D


C     The following code is a substantially modified version of subroutine
C     FLIMIT [Finite-Difference Techniques for Vectorized Fluid Dynamics
C     Calculations, D. L. Book (ed.), pp. 171, Springer-Verlag,
C     New York, 1981] which implements Zalesak's multidimensional flux
C     limiter [Zalesak, J. Comp. Phys., vol. 31, pp. 335, 1979]

C     PRELIMIT, HI_LIMIT and LO_LIMIT together comprise a FORTRAN subroutine
C     which implements FLIMIT in a 3-D cartesian geometry. The subroutine is
C     broken into three parts to lend some flexibility to its application
C     (normally they are called one after the other). It incorporates a tweak
C     to the flux limiter that reduces some spurious fluxes (when PRLIM =
C     .TRUE.) and the ability to look back to the previous timestep for upper
C     and lower bounds on the new solution (when FOLD = .TRUE.) as options.

C     Modifications:
C        Bug fix                                          - Oct '89
C        Tidying                                          - Jun '90
C        2-D to 3-D                                       - Sep '90
C        Full implementation of flux-zeroing prelimiter - Jan '92
C        Moved boundary conditions to main program        - Feb '92
C        Improved speed and memory use on IBM RS/6000     - Feb '92
C        Broken into blocks using the ENTRY statement     - Mar '92

C     Calling sequence:

C     FLX, FLY, FLZ    - Raw (unlimited) antidiffusive fluxes. Dimensionally,
C                        the fluxes should have the same units as FTD (below)
C                        multiplied by a volume. They must be real fluxes (like
```

```
C                       grams for eg.). They are corrected and returned for
C                       use in the calling program.
C                       FLX(I,J,K) - centred between FTD(I,J,K) & FTD(I+1,J,K)
C                                  - defined for I=1,NXM1 J=1,NY K=1,NZ
C                       FLY(I,J,K) - centred between FTD(I,J,K) & FTD(I,J+1,K)
C                                  - defined for I=1,NX J=1,NYM1 K=1,NZ
C                       FLZ(I,J,K) - centred between FTD(I,J,K) & FTD(I,J,K+1)
C                                  - defined for I=1,NX J=1,NY K=1,NZM1

C     FTD               - Array containing the time-advanced, low order
C                         ("transported and diffused") solution.

C     NX, NY, NZ        - Dimensions of mesh (restricted at compile time
C                         by variable dimensioning to NX <= NNX, NY <= NNY and
C                         NZ <= NNZ, where NNX, NNY and NNZ are defined in
C                         a PARAMETER statement).

C     VOL               - Volume of the cell centred at grid point (I,J,K)
C                         calculated by DX*DY*DZ. In order to speed things up
C                         and reduce storage requirements, the option of having
C                         a variable mesh has been removed.

C     FAA               - Array containing the solution from the previous
C                         timestep (used only if FOLD = .TRUE.).

C     IVAR              - Identifying number of variable being processed
C                         (for use in external boundary condition subroutine).


C     The COMMON block /LIMIT/ contains two scalar logical variables
C     which may be set by the user from outside the subroutine:

C     PRLIM                 Setting PRLIM = .TRUE. zeroes the flux using the
C                           criteria given in equation (14') in [Zalesak, 1979].

C     FOLD                  Setting FOLD = .TRUE. allows the limiter to look back
C                           to the solution from the previous timestep (which
C                           must be stored in array FAA) to find upper and
C                           lower bounds on the new solution.

C     The most conservative (i.e., most diffusive) choice is obtained by
C     setting PRLIM = .FALSE. and FOLD = .FALSE.. This is the recommended
C     choice for a first attempt.


      SUBROUTINE PRELIMIT (FLX, FLY, FLZ, FTD, NX, NY, NZ)

      INTEGER NNX, NNY, NNZ, NX, NY, NZ, IVAR
      PARAMETER (NNX = 22, NNY = 22, NNZ = 22)
      REAL*8 FLX(NNX,NNY,NNZ), FLY(NNX,NNY,NNZ), FLZ(NNX,NNY,NNZ),
     2       FTD(NNX,NNY,NNZ), VOL, FAA(NNX,NNY,NNZ)
      LOGICAL FOLD, PRLIM
      COMMON /LIMIT/ PRLIM, FOLD
      INTEGER I, J, K, NXM1, NYM1, NZM1, NXM2, NYM2, NZM2
```

```
      REAL*8 SA, SB, SC(NNX,NNY,NNZ), SD(NNX,NNY,NNZ)

D     PRINT *, '-> PRELIMIT'

      NXM1 = NX - 1
      NYM1 = NY - 1
      NZM1 = NZ - 1
      NXM2 = NX - 2
      NYM2 = NY - 2
      NZM2 = NZ - 2

      IF ( PRLIM ) THEN
C       Eliminate down-gradient antidiffusive fluxes...
        DO 100 K = 1, NZ
          DO 100 J = 1, NY
            DO 100 I = 2, NXM2
              SA = FLX(I,J,K)*(FTD(I+1,J,K) - FTD(I,J,K))
              SB = MIN(FLX(I,J,K)*(FTD(I,J,K) - FTD(I-1,J,K)),
     2                 FLX(I,J,K)*(FTD(I+2,J,K) - FTD(I+1,J,K)))
  100         FLX(I,J,K) = FLX(I,J,K)*MAX(0.D0, SIGN(1.D0, MAX(SA, SB)))
        DO 103 K = 1, NZ
          DO 103 J = 2, NYM2
            DO 103 I = 1, NX
              SA = FLY(I,J,K)*(FTD(I,J+1,K) - FTD(I,J,K))
              SB = MIN(FLY(I,J,K)*(FTD(I,J,K) - FTD(I,J-1,K)),
     2                 FLY(I,J,K)*(FTD(I,J+2,K) - FTD(I,J+1,K)))
  103         FLY(I,J,K) = FLY(I,J,K)*MAX(0.D0, SIGN(1.D0, MAX(SA, SB)))
        DO 106 K = 2, NZM2
          DO 106 J = 1, NY
            DO 106 I = 1, NX
              SA = FLZ(I,J,K)*(FTD(I,J,K+1) - FTD(I,J,K))
              SB = MIN(FLZ(I,J,K)*(FTD(I,J,K) - FTD(I,J,K-1)),
     2                 FLZ(I,J,K)*(FTD(I,J,K+2) - FTD(I,J,K+1)))
  106         FLZ(I,J,K) = FLZ(I,J,K)*MAX(0.D0, SIGN(1.D0, MAX(SA, SB)))
      END IF

D     PRINT *, '<- PRELIMIT'
      RETURN


      ENTRY HI_LIMIT (FLX, FLY, FLZ, FTD, VOL, FAA, IVAR)

D     PRINT *, '-> HI_LIMIT'

      IF ( FOLD ) THEN
C       Use solution from previous timestep in limiter...
        DO 200 K = 1, NZ
          DO 200 J = 1, NY
            DO 200 I = 1, NX
  200         SC(I,J,K) = MAX(FTD(I,J,K), FAA(I,J,K))
      ELSE
C       Don't use previous solution in limiter...
        DO 203 K = 1, NZ
          DO 203 J = 1, NY
```

```
          DO 203 I = 1, NX
  203       SC(I,J,K) = FTD(I,J,K)
      END IF

      DO 210 K = 2, NZM1
        DO 210 J = 2, NYM1
          DO 210 I = 2, NXM1
C           Set limits on antidiffusive flux...
  210       SD(I,J,K) = (0.5D0*MAX(SC(I-1,J,K) + SC(I,J,K),
     2                             SC(I+1,J,K) + SC(I,J,K),
     3                             SC(I,J-1,K) + SC(I,J,K),
     4                             SC(I,J+1,K) + SC(I,J,K),
     5                             SC(I,J,K-1) + SC(I,J,K),
     6                             SC(I,J,K+1) + SC(I,J,K),
     7                             2.0D0*SC(I,J,K))
     8                  - FTD(I,J,K))*VOL
      DO 220 K = 2, NZM1
        DO 220 J = 2, NYM1
          DO 220 I = 2, NXM1
C           Calculate antidiffusive flux into cell (x-component)...
  220       SC(I,J,K) = MAX(0.D0, FLX(I-1,J,K)) - MIN(0.D0, FLX(I,J,K))
      DO 223 K = 2, NZM1
        DO 223 J = 2, NYM1
          DO 223 I = 2, NXM1
C           Calculate antidiffusive flux into cell (y-component)...
  223       SC(I,J,K) = SC(I,J,K) +
     2                  MAX(0.D0, FLY(I,J-1,K)) - MIN(0.D0, FLY(I,J,K))
      DO 226 K = 2, NZM1
        DO 226 J = 2, NYM1
          DO 226 I = 2, NXM1
C           Calculate antidiffusive flux into cell (z-component)...
  226       SC(I,J,K) = SC(I,J,K) +
     2                  MAX(0.D0, FLZ(I,J,K-1)) - MIN(0.D0, FLZ(I,J,K))
      DO 230 K = 2, NZM1
        DO 230 J = 2, NYM1
          DO 230 I = 2, NXM1
C           Calculate preliminary limiting factor...
            IF ( SC(I,J,K) .LE. SD(I,J,K) ) THEN
              SD(I,J,K) = 1.D0
            ELSE IF ( SC(I,J,K) .EQ. 0.D0 ) THEN
              SD(I,J,K) = 0.D0
            ELSE
              SD(I,J,K) = SD(I,J,K)/SC(I,J,K)
            END IF
  230       CONTINUE

      CALL BCOND (SD, SD, IVAR)

      DO 240 K = 1, NZ
        DO 240 J = 1, NY
          DO 240 I = 1, NXM1
C           Calculate final limiting factor and correct antidiffusive flux...
  240       FLX(I,J,K) = FLX(I,J,K)*MAX(-SIGN(SD(I,J,K), FLX(I,J,K)),
     2                                SIGN(SD(I+1,J,K), FLX(I,J,K)))
```

```
      DO 243 K = 1, NZ
        DO 243 J = 1, NYM1
          DO 243 I = 1, NX
C           Calculate final limiting factor and correct antidiffusive flux...
  243       FLY(I,J,K) = FLY(I,J,K)*MAX(-SIGN(SD(I,J,K), FLY(I,J,K)),
     2                                 SIGN(SD(I,J+1,K), FLY(I,J,K)))
      DO 246 K = 1, NZM1
        DO 246 J = 1, NY
          DO 246 I = 1, NX
C           Calculate final limiting factor and correct antidiffusive flux...
  246       FLZ(I,J,K) = FLZ(I,J,K)*MAX(-SIGN(SD(I,J,K), FLZ(I,J,K)),
     2                                 SIGN(SD(I,J,K+1), FLZ(I,J,K)))

D     PRINT *, '<- HI_LIMIT'
      RETURN


      ENTRY LO_LIMIT (FLX, FLY, FLZ, FTD, VOL, FAA, IVAR)

D     PRINT *, '-> LO_LIMIT'

      IF ( FOLD ) THEN
        DO 300 K = 1, NZ
          DO 300 J = 1, NY
            DO 300 I = 1, NX
  300         SC(I,J,K) = MIN(FTD(I,J,K), FAA(I,J,K))
      ELSE
        DO 303 K = 1, NZ
          DO 303 J = 1, NY
            DO 303 I = 1, NX
  303         SC(I,J,K) = FTD(I,J,K)
      END IF

      DO 310 K = 2, NZM1
        DO 310 J = 2, NYM1
          DO 310 I = 2, NXM1
  310       SD(I,J,K) = (FTD(I,J,K) -
     2                  0.5D0*MIN(SC(I-1,J,K) + SC(I,J,K),
     3                            SC(I+1,J,K) + SC(I,J,K),
     4                            SC(I,J-1,K) + SC(I,J,K),
     5                            SC(I,J+1,K) + SC(I,J,K),
     6                            SC(I,J,K-1) + SC(I,J,K),
     7                            SC(I,J,K+1) + SC(I,J,K),
     8                            2.0D0*SC(I,J,K)))*VOL
      DO 320 K = 2, NZM1
        DO 320 J = 2, NYM1
          DO 320 I = 2, NXM1
  320       SC(I,J,K) = MAX(0.D0, FLX(I,J,K)) - MIN(0.D0, FLX(I-1,J,K))
      DO 323 K = 2, NZM1
        DO 323 J = 2, NYM1
          DO 323 I = 2, NXM1
  323       SC(I,J,K) = SC(I,J,K) +
     2                 MAX(0.D0, FLY(I,J,K)) - MIN(0.D0, FLY(I,J-1,K))
      DO 326 K = 2, NZM1
```

```
      DO 326 J = 2, NYM1
        DO 326 I = 2, NXM1
326       SC(I,J,K) = SC(I,J,K) +
     2              MAX(0.D0, FLZ(I,J,K)) - MIN(0.D0, FLZ(I,J,K-1))
      DO 330 K = 2, NZM1
        DO 330 J = 2, NYM1
          DO 330 I = 2, NXM1
            IF ( SC(I,J,K) .LE. SD(I,J,K) ) THEN
              SD(I,J,K) = 1.D0
            ELSE IF ( SC(I,J,K) .EQ. 0.D0 ) THEN
              SD(I,J,K) = 0.D0
            ELSE
              SD(I,J,K) = SD(I,J,K)/SC(I,J,K)
            END IF
330         CONTINUE

      CALL BCOND (SD, SD, IVAR)

      DO 340 K = 1, NZ
        DO 340 J = 1, NY
          DO 340 I = 1, NXM1
340         FLX(I,J,K) = FLX(I,J,K)*MAX(SIGN(SD(I,J,K), FLX(I,J,K)),
     2                               -SIGN(SD(I+1,J,K), FLX(I,J,K)))
      DO 343 K = 1, NZ
        DO 343 J = 1, NYM1
          DO 343 I = 1, NX
343         FLY(I,J,K) = FLY(I,J,K)*MAX(SIGN(SD(I,J,K), FLY(I,J,K)),
     2                               -SIGN(SD(I,J+1,K), FLY(I,J,K)))
      DO 346 K = 1, NZM1
        DO 346 J = 1, NY
          DO 346 I = 1, NX
346         FLZ(I,J,K) = FLZ(I,J,K)*MAX(SIGN(SD(I,J,K), FLZ(I,J,K)),
     2                               -SIGN(SD(I,J,K+1), FLZ(I,J,K)))

D     PRINT *, '<- LO_LIMIT'
      RETURN

      END
```

Contents of file array_subs.dat:

```
  RHO at step number     98
K=14        2             3             4             5             6
 6     2.6058944     2.6067593     2.6072901     2.6060160     2.6039584
 5     2.6041607     2.6055345     2.6073553     2.6077075     2.6059805
 4     2.6041436     2.6041941     2.6058104     2.6073543     2.6073008
 3     2.6041377     2.6041538     2.6041946     2.6055413     2.6067488
 2     2.6041352     2.6041378     2.6041435     2.6041652     2.6058940

  RVX at step number     98
K=14        2             3             4             5             6
```

6	-9.68341326E-04	-2.37242858E-03	-2.81210680E-03	-2.01058194E-03	-6.28009233E-04
5	-5.97311878E-05	-1.13347635E-03	-2.26227191E-03	-2.67035978E-03	-1.59182340E-03
4	6.69537479E-05	-1.07664054E-04	-1.04655122E-03	-2.23370273E-03	-2.18767812E-03
3	1.73276104E-05	1.09668048E-05	-3.74160941E-05	-8.37666216E-04	-1.57072556E-03
2	9.87263778E-06	1.55672083E-05	1.09534899E-05	-3.57702897E-05	-7.28720199E-04

RVY at step number 98

K=14	2	3	4	5	6
6	-7.22936542E-04	-1.56882864E-03	-2.18864181E-03	-1.60347910E-03	-6.37215412E-04
5	-3.41697276E-05	-8.35070996E-04	-2.23197203E-03	-2.67742988E-03	-2.01100766E-03
4	1.10006529E-05	-3.52271503E-05	-1.04507625E-03	-2.25988261E-03	-2.81490421E-03
3	1.57886367E-05	1.11236365E-05	-1.07621633E-04	-1.13058346E-03	-2.37845593E-03
2	9.83132879E-06	1.73806029E-05	6.71253841E-05	-6.39656574E-05	-9.64717215E-04

RVZ at step number 98

K=14	2	3	4	5	6
6	3.40608902E-04	5.67204987E-04	6.66519950E-04	4.06719583E-04	-8.61510584E-05
5	-1.60304022E-05	3.39029754E-04	6.95448552E-04	7.28319981E-04	4.12297513E-04
4	-7.94434777E-06	-1.55042186E-05	3.60884336E-04	6.98779884E-04	6.68178515E-04
3	-4.02303741E-06	-6.70879942E-06	-1.58742287E-05	3.39021035E-04	5.66098292E-04
2	-3.23785200E-06	-4.14460403E-06	-8.60620342E-06	-1.68642277E-05	3.40501701E-04

S at step number 98

K=14	2	3	4	5	6
6	1.0011064	1.0016608	1.0020065	1.0011942	.99986541
5	.99999655	1.0008874	1.0020164	1.0022651	1.0011844
4	.99998528	1.0000221	1.0010606	1.0020246	1.0020045
3	.99998147	.99999184	1.0000224	1.0008912	1.0016538
2	.99997984	.99998152	.99998521	.99999933	1.0011058

BX at step number 98

K=14	2	3	4	5	6
6	1.68035446E-04	3.87664857E-04	4.53123984E-04	4.02399601E-04	1.71381275E-04
5	-7.19606779E-06	1.52645026E-04	3.26837541E-04	3.84748525E-04	2.71365500E-04
4	-1.36747117E-05	-9.21373378E-06	1.35935365E-04	3.11649260E-04	2.93835305E-04
3	-2.16588716E-06	-3.93793001E-06	-7.10006136E-06	1.10911790E-04	1.92962693E-04
2	-3.10797831E-07	-1.53251294E-06	-4.24102759E-06	-8.38391307E-06	8.78128165E-05

BY at step number 98

K=14	2	3	4	5	6
6	8.81277820E-05	1.92430344E-04	2.93876841E-04	2.68444810E-04	1.65144920E-04
5	-7.10417599E-06	1.10219529E-04	3.11144796E-04	3.85671363E-04	3.96821748E-04
4	-3.85168675E-06	-7.07834017E-06	1.36007284E-04	3.29371849E-04	4.49548433E-04
3	-1.60655087E-06	-3.94603104E-06	-9.16420322E-06	1.52918859E-04	3.89996158E-04
2	-3.17562587E-07	-2.14036296E-06	-1.36189803E-05	-7.33077992E-06	1.68364767E-04

BZ at step number 98

K=14	2	3	4	5	6
6	.96880628	.96918127	.96943540	.96913802	.96861869
5	.96824485	.96876763	.96931533	.96953955	.96913296
4	.96823742	.96828156	.96885740	.96931412	.96942995
3	.96823469	.96824194	.96828184	.96877181	.96918137
2	.96823372	.96823471	.96823741	.96824538	.96880874